

Keywords Extraction Using Page Rank Algorithm for Arabic Text

BY

Meran Mohammad Abed Al Rahman Al Hadidi

Supervisor

Dr. Hasan Muaidi AlSerhan

Assistant Professor

Co-Supervisor

Dr. Muath Refat Al Zghool

Assistant Professor

Submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science

Faculty of Graduate Studies at AL-Balqa' Applied University

Salt-Jordan

1, August, 2013

Declaration of Authorship/Originality

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledgment within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and in the preparation of the thesis itself has been acknowledged. I certify that all information sources and literature used are indicated in the thesis.

Signiture of candidate

....

Committe Decision

This thesis was successfully defended on 1/8/2013.

Examination Commitee	**Signiture**
Dr. Hasan Muaidi Al Serhan, Chairman Assistant Professor, Artificial Intelligence	....
Dr. Muath Refaat Al Zghool, Co-Supervisor Assistant Professor, Information Retrieval	....
Dr. Mohammad F. Ababneh, Member Associate Professor, Virtual Reality	....
Dr. Shihadeh F. Alqrainy, Member Assitant Professor, Artificial Intelligence	....
Dr. Qasem A. Al-Radaideh, External Examiner Associate Professor, Data Mining Yarmouk University	....

Dedication

I lovingly dedicate this thesis to my parents and my husband, who supported and encouraged me each step of the way.

Meran

Acknowledgment

First and foremost, I have to thank ALLAH for giving me the strength and the health to complete this thesis.

I would like to sincerely thank my supervisors, Dr. Hasan Muaidi Al Serhan and Dr. Muath Refat Al Zghool for their guidance and support throughout this study.

I would also like to thank my parents and my husband for their love and support throughout my life, and for their kind co-operation and encouragement which help me in completion of this thesis

I also thank my colleagues at Al Balqa' Applied University for sharing experiences and knowledge during the time of study.

Finally, to all my friends, thank you for your understanding and encouragement in my many moments of crisis. Your friendship makes my life a wonderful experience. I cannot list all the names here, but you are always on my mind.

Contents

List of Tables

List of Figures



Abstract

Keywords Extraction Using Page Rank Algorithm for Arabic Text

By

Meran Mohammad Abed Al Rahman Al Hadidi

Supervisor

Dr. Hasan Muaidi AlSerhan

Assistant Professor

Co-Supervisor

Dr. Muath Refat Al Zghool

Assistant Professor

The main aim of this thesis is to extract keywords and keyphrases from Arabic text using page rank algorithm. Keywords can be defined as the most informative and important words that represent the topic of the whole text to help people to determine what the article is primarily talking about without reading it. Keywords extraction is an important technology in many areas of information technology and document processing, such as document tagging, text clustering, text categorization, text summarization, and text retrieval. Many keywords extraction algorithms have been implemented, most of the work in this area was carried out for the English text; on the other hand few researches have been carried out for the Arabic text. This is due to that Arabic language is morphologically complex and it has much richer morphology than English.

In this research, keywords and keyphrases are extracted by constructing a graph for a given research article's abstract with its title using candidate words as nodes, and co-occurrence relation to draw edges between them within a specified window size. Then runs the page rank algorithm upon the graph to rank each keyword's importance. Vertices are sorted by their page rank scores in descending order and a rate of tokens are chosen as keywords. Each keyword is expanded into a set of keyphrases by searching for each occurrence of the token in the original text, and for each occurrence collecting all adjacent keywords and concatenating them into a keyphrase.

Several experiments were applied on a dataset that consists of 100 documents for training and 50 documents for testing set, then the results were evaluated using precision, recall, and F-measure. The maximum recall that can be achieved on the testing set of this collection which consists of 50 Arabic documents is 63%. Since not all the manual keywords and keyphrases are existed in the articles' abstracts and their titles. This proposed system achieved 25% of recall which is acceptable and competitive compared with the recall value that has been achieved on English testing collection which consists of 500 English documents by Mihalcea and Tarau, they have achieved 42% of recall where the ideal recall of their testing set was 78%. Despite of the difficulties and challenges of the Arabic language and using less number of documents in the Arabic testing collection than English. Keywords and keyphrases extraction system using page rank algorithm works well because it does not only rely on the local context of a word in a text (vertex), but rather it takes into account information recursively drawn from the entire text (graph).

Chapter 1

INTRODUCTION

1.1 Overview

Arabic is one of the greatest and the most widespread languages in the world. Its graceful script, magnificent style and rich vocabulary give the language a unique character and flavour. It is an official language in 22 countries. It ranks the sixth in the world's league languages table, with an estimated 186 million native speakers. Being the language of the Quran, Arabic is highly respected across the Muslim world. Many non-Arab muslim children begin learning Arabic at early age, to enable them to read and understand the Quran (Muaidi, 2008).

The Arabic language has 28 characters written from right to left. It is a semitic language characterized by a wide number of linguistic varieties. The characterisic feature of Semitic languages is their basis of consonantal roots, mostly triliteral (three-lettered). It has three types of dialects: classical, modern standard, colloquial Arabic dialects (Belkredim & El-Sebai, 2009).

To generate Arabic text, an Arabic grammar is needed. Arabic grammar has two categories: morphology and syntax. Morphology studies the forms of words and their transformations to intended meanings. Syntax studies the case endings of words and their positions in the sentence. An Arabic sentence consists of words. Arabic words are divided into three types as shown in Figure 1.1: Noun, verb and particle (Al-Hamad & Al-Zoubi, 1993).

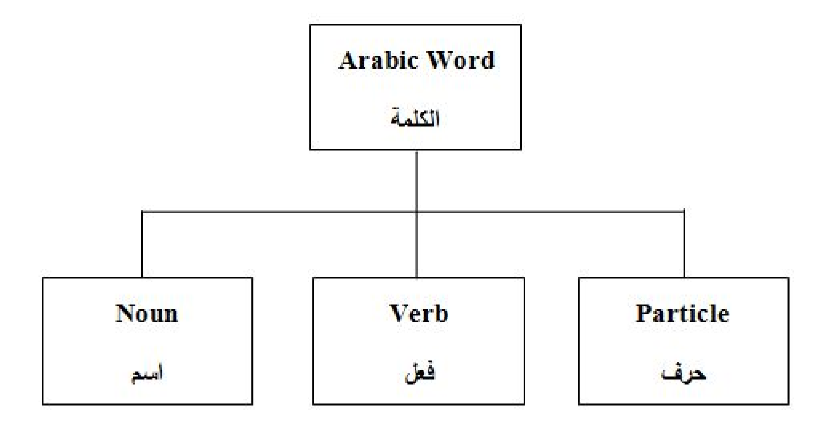


Figure 1.1: Structure of Arabic Word (Al-Hamad & Al-Zoubi, 1993)

Noun is a token that has a meaning in itself without being connected with time. Verb is a token that indicates a state or a fact happening in the past, present, or future. Particle may consist actually of more than one letter. Particles can be used in meanings of the following types: introduction, exclusion, restriction, inauguration, interrogation, future, rectification, imperative, stimulation, authenticity, selection, solicitation, similitude, variability, astonishment, definition, causality, interpretation, separation, paucity, profusion, wish, premonition, regret, confirmation, answer, rejection, augmentation, condition, circumstance, exposition, attraction, finality, oath, originality, surprise, lamentation, call, negation, or interdiction. Particles are used in sentence construction (Al-Muhtaseb & Mellish, 1998).

Natural Language Processing (NLP) is the ability of a computer program to understand human speech as it is spoken. It is a technique where computers can become more human by reducing the distance between humans and computers. They are used to understand and manipulate natural language text or speech. Information retrieval is an area where applications of natural language processing can be seen to extract information required from a large database. There are lots of places where this technique is applied to get things that are needed faster. One of its applications is Keywords Extraction System which is the topic of this research (Mihalcea et al., 2006).

Keywords and keyphrases provide a simple way of describing a document, giving the reader some clues about its contents. They can be useful in various applications such as retrieval engines, browsing interfaces, thesaurus construction or text mining. A list of extracted keywords and keyphrases associated with a document may serve as indicative summary or document metadata, which helps readers in searching relevant information. Manual selection of keywords and keyphrases from a document by a human is not a ran-

dom act. Hence, automatic keywords and keyphrase extraction is not a trivial task and it needs to automated due to its usability in managing information (Sarkar et al., 2010).

1.2 Problem Statement

Keywords or keyphrases extraction is the task to identify a small set of words or phrases from a document that can best describe the document. For example, academic papers are often accompanied by a set of keyphrases freely chosen by the author. In libraries professional indexers select keyphrases from a controlled vocabulary according to defined cataloguing rules. On the Internet, digital libraries to organize and provide a thematic access to their data.

Automatic keywords extraction should be done systematically and with either minimal or no human intervention, depending on the model. The goal of automatic extraction is to apply the power and speed of computation to the problems of access and discoverability. Adding value to information retrieval without the significant costs and drawbacks associated with human indexers (Hulth, 2003b).

The idea of using the graph-based approach for information retrieval systems appeared in the early days of the research. The idea of page rank is to utilize the hyperlink structure of Web documents, in addition to their contents, to rank retrieved documents (Lawrence et al., 1998). Page rank constructs a graph structure, where edges represent hyperlinks, and nodes represent web documents. It then assigns higher weights to nodes with more edges coming from other nodes. Thus, it effectively collects "Votes" from web pages to rank the pages. Mihalcea and Tarau (Mihalcea & Tarau, 2004) have proposed text rank for English texts. A graph-based ranking algorithm similar to page rank, for extracting keywords and keyphrases from documents. With text rank, nodes in the graph represent words instead of web documents, and edges represent word co-occurrences within a specified window size instead of hyperlinks.

Based on this idea (Text Rank), keywords extraction for Arabic texts is implemented in this research, despite of Arabic language difficulties and differences from English language.

1.3 Research Questions and Objectives

Many keywords extraction algorithms have been implemented, most of the work in this area was carried out for the English text. On the other hand, very few researches have been carried out for the Arabic text. The nature of Arabic text is different than the

English text and the preprocessing of the Arabic text is difficult and more challenging.

Several challenges complicated the keywords extraction in Arabic language. Compared to the large number of publications and available resources and lexicons in English, the Arabic keywords extraction is still immature and has less number of publications and very few resources. The lack of resources, the variant sources of ambiguity, and rich metaphoric script usage remain the most challenging problems for Arabic NLP researchers. Arabic language is a highly inflected language; it has much richer morphology than English. Arabic words have two genders, feminine and masculine; three numbers, singular, dual, and plural; and three grammatical cases, nominative, accusative, and genitive (Kouninef & AL-Johar, 2011).

Keywods extraction using page rank algorithm was implemented on English text by Rada Mihalcea and Paul Tarau (Mihalcea & Tarau, 2004). They introduced text rank, a graph-based ranking model for text processing. They showed how this model can be successfully in natural language applications, especially in keywords extraction.

The research questions of this study are as follows:

- Is it possible to extract keywords from Arabic texts using page rank algorithm as from English texts despite of Arabic Language difficulties?.
- Are the linguistic features such as part of speech help to improve the performance of keywords and keyphrases extraction system?.
- Does the word position feature (in title or abstract) help to improve the performance of keywords and keyphrases extraction?.
- Does the word frequency feature in each document help to improve the performance of keywords and keyphrases extraction?.

1.4 Significance of the Study

The main objective of this research is to extract keywords and keyphrases from Arabic documents using page rank algorithm. Keywords in a document provide important information about the content of the document. They can help users search through information more efficiently or decide whether to read a document. They can also be used for a variety of language processing tasks such as text categorization and information retrieval. Keywords search is a usable and powerful tool which enables efficient scanning of large document collections.

The manual extraction of keywords is slow, expensive and full with mistakes. Therefore, most algorithms and systems to help people perform automatic keyword extraction have been proposed. Keywords extraction using page rank algorithm was applied on English language. It works well because it does not only rely on the local context of a text unit (vertex), but rather it takes into account information recursively drawn from the entire text (graph) (Mihalcea & Tarau, 2004).

The main objectives of this research are:

- To design a keywords and keyphrases extraction system using page rank algorithm for a collection of 150 Arabic documents (abstracts and their titles) regarding the linguistic and the word position feature in the document (abstract or title). And to compare the generated automatic keywords with a dataset of keywords that were found manually.
- To suggest more meaningful and expressive generated keywords and keyphrases for Arabic documents (abstracts and their titles) than the manual ones.

1.5 Contributions

The thesis contributions are summarized as follows:

- A new dataset that is composed of 150 Arabic documents (abstracts and their titles) is concluded and implemented.
- Arabic words in the collected dataset documents (abstracts and their titles) were tagged into three main categories: Nouns, verbs and particles.
- A new keywords and keyphrases extraction system using page rank algorithm has been designed and implemented for the previous collected Arabic dataset. The proposed system in this research is different from Mihalcea and Tarau system in terms of:
  - Applying the proposed system on Arabic documents (abstracts and titles).
  - Considering the word position in the documnet either in abstract or title.
  - Using word frequency in the document.
- An evaluation system has been constructed to compare the generated automatic keywords and keyphrases with the manual ones for the Arabic dataset.

1.6 Thesis Structure

This thesis is divided into seven chapters. Chapter 1, the present chapter is the introduction chapter. The remaining chapters are organized as follows:

- **Chapter 2:** This chapter shows a number of important topics in Arabic language. It starts with a breif history of Arabic language. Arabic alphabets are clarified with their forms depending on their locations in words. The transliteration is explained. Arabic syntax is described with its three types: Nouns, verbs and particles. In addition to many examples that illustrate these types.

- **Chapter 3:** This chapter shows many important topics starting from an explanation to keywords extraction system. A detailed explanation of page rank algorithm and a clear example are added. Text rank model that simulates the same idea of page rank, using words instead of web pages is explained. Where text is implemented as a graph (Forward, Backward, Bidirectional or weighted) on the collected dataset that contains 150 Arabic documents from different disciplines. Finally the evaluation methods were explained.

- **Chapter 4:** This chapter details the history of keywords and keyphrases extraction systems and the previous works done in this field. All approaches in keywords extraction are viewed either they were the main subjects or stages in other searches such as text summarization, clustering or identification. Keywords and keyphrases extraction systems on English, Arabic and other language are mentioned.

- **Chapter 5:** This chapter is concerned with the thesis methodology and implementation of the keywords extraction system using page rank algorithm. It starts by explaining the basic steps of the algorithm implementation using forward, backward and bidirectional graphs and the architecture of the basic system. Additional steps added to the basic system implementation such as stopwords removal and words tagging process are also clarified. All the rules that are used in the proposed system are listed and discussed. The features of the system are mentioned.

- **Chapter 6:** This chapter details the experiments results of keywords and keyphrases extraction system using text rank adapted from Mihalcea and Tarau for Arabic dataset. Many experiments with different window sizes were implemented on the training set to choose the best one according to the highest value of F-Measure. Then apply it on the testing set to be adopted as the final results.

- **Chapter 7:** This chapter presents the conclusions of this study. A summary of the thesis is introduced. The research question and the contributions of this study are discussed. Finally future works on the current system are presented and suggested.

Chapter 2

The Arabic Language

2.1 Introduction

This chapter shows a number of important topics of Arabic language. It begins with a brief history of Arabic language. Explanation of Arabic alphabets and their forms in the words. The Transliteration is defined. Finally, Arabic syntax is clarified into: Nouns, verbs and particles with examples.

2.2 The History of Arabic Language

Arabic is the official language of many countries in the Middle East such as Egypt, Iraq, Jordan, Lebanon, Libya, Morocco, Saudi Arabia and Sudan. It is one of the six official languages of the United Nations. Arabic language is a very rich language with complex morphology, so it has a very different and difficult structure than other languages. By the 7^{th} Century A.D., Arabic started to spread to the Middle East as many people started to convert to Islam. During this time of religious conversions, Arabic replaced many South Arabian languages. Most of which are no longer commonly spoken or understood languages (Muaidi, 2008).

Arabic belongs to the Semitic family of languages. It is widely spoken on two continents, from North Africa to the Arabian Peninsula. The Arabic language was, and still is, easily capable of creating new words and terminology in order to adapt to the demands of new scientific and technological discoveries. The most important thing to know about the Arabic language is that, like other Semitic languages, it is based on what is usually called a "consonantal root system". Which means that almost every word in the language is ultimately derived from one or another "root" usually a verb. This root almost always consists of three letters. By making changes to the root letters, adding a letter to the beginning of the root, changing vowels between the consonants, or inserting ex-

tra consonants new words with new meanings are produced (Belkredim & El-Sebai, 2009).

Arabic Language dialects has three types: Classical Arabic (العربية الفصحى), modern standard Arabic (العربية الحديثة), and colloquial Arabic dialects (العربية العامية) (Belkredim & El-Sebai, 2009). Quranic or classical Arabic is the Arabic of Islam's holy book, the Quran (or Koran). It is very old, dating from the late 600's when the Quran was written down. It is used in the Quran and in the holy books of Islam. No one speaks classical Arabic as a native, nor is it used for conversation. It is learned primarily for reciting and reading the Quran. It could also be viewed as the language of the pre-Islamic poets. This language is rarely used in today's everyday writing (Shah, 2008).

Modern Standard Arabic (MSA) is an updated version of classical Arabic which is taught in the schools of Arab countries. It is the language of today's Arabic newspapers, magazines, periodicals, letters, modern writers, modern literature and education. No one speaks it as a native language but it is used as a common language for people who speak very different varieties of Arabic or by second-language speakers. It is also used as the medium of oral communication in formal speeches and in television and radio broadcasts. MSA could be viewed as classical, however, MSA differs from classical Arabic by adopting minor stylistic changes and expanding the lexicon to include new technical terms (Belkredim & El-Sebai, 2009).

Colloquial Arabic dialects, on the other hand, consist of the languages of the different Arab countries. They are used for everyday oral communications by the people of different areas. Arabs use the colloquial language in most informal settings and in all their daily activities and connections; but these dialects are seldom written for official documents. So the modern standard Arabic is the suitable choice for formal settings. There are no written transcripts for such dialects. In this research, keywords or keyphrases extraction system using page rank algorithm will be implemented on Arabic texts written in classical or MSA (Belkredim & El-Sebai, 2009).

2.3 Arabic Alphabets

The Arabic alphabet or Arabic abjad is the Arabic script as it is codified for writing the Arabic language. It was derived from a type of Aramaic via the Nabatean cursive alphabet. By the earliest known document dating from 512 AD. The Aramaic language has fewer consonants than Arabic. New letters were created around the 7th century by adding dots to existing letters (Muaidi, 2008).

Arabic Language includes 28 letters and it is written from right to left, in a cursive

style. Which is to say that Arabic cannot be written with unconnected, separated letters as English, therefore all letters must be connected together in general. For example the word AlMadrasah [المدرسة] that corresponds the word School in English written in a cursive style with all its letters being connected. Three letters from the Arabic letters appear in different shapes (Muaidi, 2008):

- Hamza [ء], where it can be on Alef letter [أ], below [إ], also it can be on Waw letter [ؤ], or Alef Maqsura [ئ].
- Taa' Mrboota [ة ـة], which is a special case of the letter Tee that appears always at the end of the word [ت] as in AlMadrasah [المدرسة].
- Alef Maqsura [ى], which is a special case of letter Alef word [ا] that appears always at the end of the word as in Fata [فتى], a boy in English.

There are no capital and small letters in Arabic. There is no discrete handwriting in Arabic, so letters are written in four different forms depending on their location in words. This is the only aspect of Arabic that makes it look complicated. Transliteration can be defined as a method of mapping the script of one language to the best matching script of another language, word by word, or perfectly letter by letter. It is used to represent Arabic words for readers who cannot read the Arabic script. Most of the letters in Arabic have four forms: Stand alone, word initial, medial and word ending depending on its position in the word (Habash et al., 2007). Table 2.1 shows the Arabic alphabets and their four forms depending on their positions in words with their transliterations.

Table 2.1: Arabic Alphabets in Four Different Positions with their Transliterations

Alphabet	Initial Form	Middle Form	End Form	Alone	Transliteration
ألف	ا	ـا	ـا	ا	Alef
باء	بـ	ـبـ	ـب	ب	Baa
تاء	تـ	ـتـ	ـت	ت	Taa
ثاء	ثـ	ـثـ	ـث	ث	Thaa
جيم	جـ	ـجـ	ـج	ج	Jiim
حاء	حـ	ـحـ	ـح	ح	Haa
خاء	خـ	ـخـ	ـخ	خ	Khaa
دال	د	ـد	ـد	د	Daal
ذال	ذ	ـذ	ـذ	ذ	Thaal
راء	ر	ـر	ـر	ر	Raa
زين	ز	ـز	ـز	ز	Zaay
سين	سـ	ـسـ	ـس	س	Siin
شين	شـ	ـشـ	ـش	ش	Shiin
صاد	صـ	ـصـ	ـص	ص	Saad
ضاد	ضـ	ـضـ	ـض	ض	Daad
طاء	طـ	ـطـ	ـط	ط	Taa
ظاء	ظـ	ـظـ	ـظ	ظ	Thaa
عين	عـ	ـعـ	ـع	ع	Ayn
غين	غـ	ـغـ	ـغ	غ	Ghayn
فاء	فـ	ـفـ	ـف	ف	Faa
قاف	قـ	ـقـ	ـق	ق	Qaaf
كاف	كـ	ـكـ	ـك	ك	Kaaf
لام	لـ	ـلـ	ـل	ل	Laam
ميم	مـ	ـمـ	ـم	م	Miim
نون	نـ	ـنـ	ـن	ن	Noon
هاء	هـ	ـهـ	ـه	ه	Haa
واو	و	ـو	ـو	و	Waaw
ياء	يـ	ـيـ	ـي	ي	Yaa

2.4 Arabic Syntax

In English language there are eight parts of speech namely (Täckström et al., 2013):

- Noun

- Pronoun
- Verb
- Adjective
- Adverb
- Preposition
- Conjunction
- Interjection

The Arabic language is made up of words, and these words are of three types. These three parts are more general and encompass all eight English language parts of Speech (El-Hadj et al., 2009). They are known as:

- Nouns
- Verbs
- Particles

Arabic nouns encompass nouns, pronouns, adjectives, adverbs and Interjections in English. for example [بيت] which is pronounced as Bayt that means a house in English. Arabic verbs such as [كسر] that is pronounced as Kasara that means broke in English. and Arabic particles which encompass prepositions and conjunctions in English, such as [إلى] (ila) meaning (to), [في] (fee) meaning (in) and [و] (wa) meaning (and) (El-Hadj et al., 2009).

2.4.1 Nouns

A noun is a word that indicates a meaning that is not associated with time and is used to name or denote a class of things, places, people, ideas, or qualities. All nouns are either singular (one only), dual (two only) or plural (more than two). Dual and plural nouns can be formed by adding suffixes to a singular noun, but some plurals are irregular in their grammatical formation (Muaidi, 2008). Table 2.2 shows examples of Arabic nouns, their singular, dual and plural forms.

Table 2.2: Examples on Arabic Nouns and their Forms

Word	Singular	Dual	Plural	Pronunciation	English Translation
جمل	جمل	جملان	جمال	Jamal	Camel
رجل	رجل	رجلان	رجال	Rajol	Man
كتاب	كتاب	كتابان	كتب	Ketab	Book
قلم	قلم	قلمان	اقلام	Kalam	Pen
يوم	يوم	يومان	ايام	Yawm	Day
درس	درس	درسان	دروس	Dars	Lesson
طالب	طالب	طالبان	طلاب	Taleb	Student
فراشه	فراشه	فراشتان	فراشات	Farashah	Butterfly
بنت	بنت	بنتان	بنات	Bent	Girl
ولد	ولد	ولدان	اولاد	Walad	Boy
طفل	طفل	طفلان	أطفال	Tefl	Baby

Nouns in Arabic language includes adjectives and pronouns. They depend on gender, they could be masculine that is usually used when referring to a male. Or feminine when referring to a female. In most cases the feminine noun is formed by adding a special character, the ta marbuta [ـة ة], to the end of the masculine noun (Muaidi, 2008). Table 2.3 shows some examples of Arabic feminine nouns and their corresponding masculine nouns in singular form.

Table 2.3: Examples on Feminine and Masculine Arabic Nouns

Word in English	Feminine Singular	Masculine Singular
Teacher	معلم	معلمة
Researcher	باحث	باحثة
Student	طالب	طالبة
Friend	صديق	صديقة
colleague	زميل	زميلة
translator	مترجم	مترجمة
writer	كاتب	كاتبة
muslim	مسلم	مسلمة

Note from the previous table that English nouns are the same for males and females, while Arabic nouns differ between males and females. Sometimes the noun used to refer to a male and the noun used to refer to a female are completely different. Such as rajol [رجل] that means a man in English and emra'a [امرأة] which means a woman. Unfortunately, not all feminine nouns end in ta marbuta. There are no constant rule to learn about (El-Hadj et al., 2009) .

Pronouns belong to nouns in Arabic language, they are used to identify something. Therefore, they will convert something from being unidentified with respect to its owner to defined and known. They could be attached to other nouns or deattached (Muaidi, 2008). Table 2.4 shows a list of some deattached Arabic pronouns. While Table 2.5 shows some examples of the attached nouns in Arabic.

Table 2.4: A List of Some Deattached Arabic Pronouns

Pronoun in English	Pronoun in Arabic	Transliteration
I	أنا	Ana
You	أنت	Anta
He	هو	Howa
She	هي	Heia
We	نحن	Nahno

Table 2.5: Examples of some Attached Pronouns in Arabic

Word in English	Arabic Word	Transliteration
My Home	بيتي	Baite
His Book	كتابه	Ketabaho
Her Book	كتابها	Ketabaho
Their Brother	أخاهم	Akhahom
Their Father	والدهم	Waledahom
Her Pen	قلمها	Kalamoha
His Pen	قلمه	Kalamoha

2.4.2 Verbs

Verbs are words which usually express an action done by something or someone. Arabic verbs have three types: Past [الماضي, Present [المضارع] and Imperative [الأمر]. Present verbs are used to describe an action in the present time, such as yaktob [يكتب] that means writes in English. Past verbs are used to describe an action that has done in the past time such as kataba [كتب] that corresponds wrote in English. Imperative verbs are commands or requests of others to do a particular act such as oktob [أكتب] that also corresponds to write (El-Hadj et al., 2009).

As in nouns, the Arabic verbs have three numbers: Singular for one as in yadros [يدرس], which means he studies in English. Dual for two as in yadrosan [يدرسان], which means they study. And plural for more than two as in yadrosoon [يدرسون], also which means

they study (El-Hadj et al., 2009). Table 2.7 shows some examples of Arabic verbs in singular, dual and plural forms.

Table 2.6: A List of Some Arabic Verbs (Past, Present and Imperative)

Past Verb	Present Verb	Imperative Verb	English Verb
كتب (Kataba)	يكتب (Yaktob)	أكتب (Oktob)	Write
درس (Darasa)	يدرس (Yadros)	أدرس (Odros)	Study
قال (Kala)	يقول (Yakool)	قل (Kol)	Say
سمح (samaha)	يسمح (Yasmah)	اسمح (Esmah)	Let
حسب (hasaba)	يحسب (Yahseb)	احسب (Ehseb)	Compute
قارن (qaarana)	يقارن (Yokaren)	قارن (Karen)	Compare
نسخ (nasakha)	ينسخ (Yansakh)	انسخ (Ensakh)	Copy
كشف (kashafa)	يكشف (Yakshef)	اكشف (Ekshef)	Detect
ضحك (daheka)	يضحك (Yadhak)	اضحك (Edhak)	Laugh
ترك (taraka)	يترك (Yatrok)	أترك (Otrok)	Leave
صرخ (sarakha)	يصرخ (Yasrokh)	أصرخ (Osrokh)	Scream
رفض (rafada)	يرفض (Yarfod)	أرفض (Orfod)	Refuse

Besides, Arabic verbs have two genders: Masculine for males and feminine for females as shown in some examples in Table 2.8.

Table 2.7: A List of Some Arabic Verbs (Singular, Dual and Plural)

English	Arabic Singular	Arabic Dual	Arabic Plural
Write	كتب (Kataba)	كتبا (Katabaa)	كتبوا (Katabo)
Eat	أكل (Akala)	أكلا (Akalaa)	أكلوا (Akaloo)
Arrive	وصل (Wasala)	وصلا (Wasalaa)	وصلوا (Wasaloo)
Visit	زار (Zara)	زارا (Zaraa)	زاروا (Zaroo)
Forget	نسي (Naseia)	نسيا (Naseiaa)	نسيوا (Naseioo)
Run	ركض (Rakada)	ركضا (Rakadaa)	ركضوا (Rakadoo)

2.4.3 Particles

Every Arabic word that does not have the signs of either a noun or a verb is a particle, harf [حرف]. Particles may consist actually of more than one letter. They may be deattached alone or attached to other words. Particles include (Muaidi, 2008):

- Prepositions [حروف الجر] such as [من، إلى، في].
- Conjunctions [حروف العطف] such as [و، ف، أو، ثم].

- Interrogative particles [حروف استفهام] such as [هل، أ].
- Vocative particle [حرف نداء] which is [يا].

Table 2.8 shows some examples of Arabic particles.

Table 2.8: A List of Some Arabic Particles

Particle in Arabic	Transliteration	Particle in English
إن	inna	It is true that
أن	anna	That
لكن	laakinna	But
كأن	ka'anna	It is like that
لعل	la'alla	It is hoped that
ليت	layta	It is wishwd that

Finally Arabic text is made up of nouns, verbs and particles. Nouns are names of things, verbs provide information, and particles complete the meaning.

Chapter 3

Theoretical Background

3.1 Introduction

This chapter shows many important topics starting from an explanation to keywords and keyphrases extraction system on English text. A detailed explanation of page rank algorithm and a clear example are clarified. Besides, text rank model that simulates the idea of page rank, using words instead of web pages is explained. Where text is implemented as a graph (Forward, Backward, Bidirectional or weighted). Finally the evaluation methods were performed and explained.

3.2 Keywords and Kephrases Extraction

Keywords and keyphrases are defined as a sequence of one or more words extracted from a document that provide important information about the content of the document to describe the meaning of it. Using these keywords, users can search through information more efficiently and decide whether to read a document or not. Also, keywords can be used for a variety of language processing tasks such as text categorization and information retrieval (Rose et al., 2010).

Extracting keywords manually is an extremely slow, difficult, full with mistakes and time consuming process, so it is almost impossible to extract keywords manually. Therefore automated algorithms that extract keywords and keyphrases from documents are important (Liu et al., 2009).

In order to extract keywords from documents, two methods can be used. The first method is the unsupervised approach, which assigns each word a score and ranks the words according to their scores in the document. The score is usually computed based on a combination of statistical and linguistic feature, including term frequency, word posi-

tion, signature, lexical chains and so on. The second method is the supervised keywords extraction approach that treats the task as a two class classification problem at the word level. Each word is represented by a vector of features, that can be adopted as the key of extraction. Features can be defined from linguistic, such as term significance or term location in a document (Witten et al., 1999).

Many keywords extraction algorithms have been implemented, most of the work in this area was carried out for the English text. On the other hand very few researches have been carried out for the Arabic text. The nature of Arabic text is different. The preprocessing of Arabic text is difficult and has more challenging than English text.

The main objective of this research is to extract keywords and keyphrases from Arabic texts using page rank algorithm. But instead of applying this algorithm on web pages, it is implemented on texts. A collection of Arabic documents (abstracts and their titles) is used by treating the words within the text as web pages. This method is called keywords extarction using page rank algorithm. Keywords extraction from documents abstracts is more widely informative than from full texts, since many documents on the Internet are not available as full-texts, but only as abstracts such as scientific articles (Mihalcea & Tarau, 2004).

3.3 Google Page Rank Algorithm

Google is a search engine owned by Google, Inc. Whose mission statement is to organize the world's information and make it universally accessible and useful. It is the largest search engine on the web. Google receives over 200 million queries each day through its various services. The heart of Google's searching software is page rank. It is a system for ranking web pages developed by Larry Page and Sergey Brinat Stanford University, to decide how much this web page is important compared to other web pages based on the authority measure. A common measure of authority is the in-degree pages which point to a specific page (Brin & Page, 1998).

A page rank algorithm implemented on a graph that is constructed from web pages as vertices, is a way of deciding the importance of a vertex (a web page) within this graph. By taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information. The basic idea in a graph-based ranking model is of voting. When one vertex (web page) links to another one, it is basically creating a vote for that other vertex. The higher the number of votes for a

vertex, the higher the importance of that vertex. Each vertex will have a score that is determined based on the votes that are cast for it, and the scores of the vertices casting these votes (Mihalcea & Tarau, 2004).

Let G(V,E) refers to the constructed graph. Where V is the set of vertices in the graph that represents the web pages, and E is the set of the edges between vertices. An edge is constructed in the graph between two verices when one vertex points to the other vertex. For a given vertex V, let In(V) represents the set of vertices that point to the vertex (V), and Out(V) represents the set of vertices that (V) points to. According to Brin and Page Equation 3.1 (Brin & Page, 1998), the score of a given vertex (V_i) is computed as follows:

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{Out(V_j)} S(V_j) \tag{3.1}$$

Where d refers to the damping factor, its value can be set between 0 and 1 (Brin & Page, 1998). It has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. The factor d is usually set to 0.85 (Brin & Page, 1998), and this value is used in this current research implementation.

Figure 3.1 illustrates an example that explains the idea of page rank algorithm. The graph in this example consists of a subset of pages (page A, page B, page C and page D) and their links as follows:

- Page A points to pages B, C and D
- Page B points to page D
- Page D points to page B and C

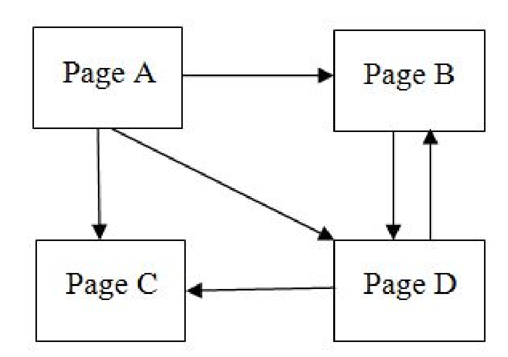


Figure 3.1: Example of Page Rank Algorithm.

Initially, the graph G=(V,E) is constructed as shown in Figure 3.1, where V = {A, B, C, D} and E = {(A,B), (A,C), (A,D), (B,D), (D,B), (D,C)}. An initial page rank value is assigned for each vertex in the graph by 1.

(S(A)=1, S(B)=1, S(C)=1 and S(D)=1)

Then Equation 3.1 is applied on each vertex for n number of iterations until convergence is achieved. Convergence is achieved when a value called the error rate for each vertex in the graph approaches or falls below a given threshold value. The error rate is computed from two successive iterations, which can be defined as the difference between the score of the vertex at iteration K+1, and the score computed at iteration k, as shown in Equation 3.2 (Mihalcea & Tarau, 2004).

$$Error_Rate = S^{k+1}(V) - S^{k}(V) \tag{3.2}$$

The page rank scores for all pages in Figure 3.1 are computed as follows:

- For Iteration 0 (Initial Iteration):
  - S(A)=1
  - S(B)=1
  - S(C)=1
  - S(D)=1
- For Iteration 1:
  - The score for page A:

$$S(A) = (1 - 0.85) + 0.85 * 0$$
$$S(A) = 0.15$$

  - The score for page B:

$$S(B) = 0.15 + 0.85(\frac{S(A)}{OUT(A)} + \frac{S(D)}{OUT(D)})$$
$$S(B) = 0.15 + 0.85(\frac{1}{3} + \frac{1}{2})$$
$$S(B) = 0.8583$$

– The score for page C:

$$S(C) = 0.15 + 0.85(\frac{S(D)}{OUT(D)} + \frac{S(A)}{OUT(A)})$$
$$S(C) = 0.15 + 0.85(\frac{1}{2} + \frac{1}{3})$$
$$S(C) = 0.8583$$

– The score for page D:

$$S(D) = 0.15 + 0.85(\frac{S(A)}{OUT(A)} + \frac{S(B)}{OUT(B)})$$
$$S(D) = 0.15 + 0.85(\frac{1}{3} + \frac{1}{1})$$
$$S(D) = 1.2833$$

Table 3.1 shows the results of page rank scores for pages A, B, C and D. These scores are calculated after running a small Perl program that has been designed in this research to calculate the page rank scores for this example. Iterations will be continued until convergence is achieved. It shows various stages of iterations. It can be noticed that page rank scores are stable after iteration 18 for each page.

Table 3.1: Results of Page Rank Scores of the above Example for 19 Iterations

Iteration No.	A	B	C	D
0	1	1	1	1
1	0.15	0.858333333	0.858333333	1.283333333
2	0.15	0.737916667	0.737916667	0.922083333
3	0.15	0.584385417	0.584385417	0.819729167
4	0.15	0.540884896	0.540884896	0.689227604
5	0.15	0.485421732	0.485421732	0.652252161
6	0.15	0.469707169	0.469707169	0.605108472
7	0.15	0.449671101	0.449671101	0.591751093
8	0.15	0.443994215	0.443994215	0.574720436
9	0.15	0.436756185	0.436756185	0.569895082
10	0.15	0.43470541	0.43470541	0.563742757
11	0.15	0.432090672	0.432090672	0.561999599
12	0.15	0.431349829	0.431349829	0.559777071
13	0.15	0.430405255	0.430405255	0.559147355
14	0.15	0.430137626	0.430137626	0.558344467
15	0.15	0.429796398	0.429796398	0.558116982
16	0.15	0.429699717	0.429699717	0.557826939
17	0.15	0.429576449	0.429576449	0.55774476
18	0.15	0.429541523	0.429541523	0.557639982

As shown in Table 3.1 applying the algorithm for n iterations until convergence achieved. Convergence is achieved when the error rate for any vertex in the graph falls below a given threshold. A threshold value of 0.0001 is used (Mihalcea & Tarau, 2004).

Table 3.2 shows the error rate values for pages (A, B, C and D) in Figure 3.1. Note that the page rank score for page A is constant at any iteration number. Its error rate value after any iteration number is 0. Since there are no pages pointing to A, indegree for A is 0.

Table 3.2: Results of Error Rate Values for Pages(A, B, C and D) in Figure 3.1

After Iteration No.	A	B	C	D
1	0.85	0.141666667	0.141666667	0.283333333
2	0	0.120416667	0.120416667	0.36125
3	0	0.15353125	0.15353125	0.102354167
4	0	0.043500521	0.043500521	0.130501563
5	0	0.055463164	0.055463164	0.036975443
6	0	0.015714563	0.015714563	0.047143689
7	0	0.020036068	0.020036068	0.013357379
8	0	0.005676886	0.005676886	0.017030658
9	0	0.00723803	0.00723803	0.004825353
10	0	0.002050775	0.002050775	0.006152325
11	0	0.002614738	0.002614738	0.001743159
12	0	0.000740842	0.000740842	0.002222527
13	0	0.000944574	0.000944574	0.000629716
14	0	0.000267629	0.000267629	0.000802888
15	0	0.000341227	0.000341227	0.000227485
16	0	0.000096681	0.000096681	0.000290043
17	0	0.000123268	0.000123268	0.000082178
18	0	0.000034926	0.000034926	0.000104778

At the final stage, pages are sorted according to their page rank scores in descending order to show the importance of each page as follows:

$$Page(D) = 0.557539231$$

$$Page(C) = 0.429456273$$

$$Page(B) = 0.429456273$$

$$Page(A) = 0.15$$

3.4 Text Rank Model

Text rank, as outlined in Mihalcea and Tarau (2004), constructs a network graph using candidate keywords as nodes, and co-occurrence to draw edges between them, and then runs the page rank algorithm upon the graph to rank each keyword's importance.

This text rank algorithm makes use of the Hulth (2003) dataset. This dataset consists of English 2000 abstracts from the Inspec database from the years 1998 to 2002 and includes articles from two disciplines: Computers and Control, and Information Technology. Keywords are assigned by a professional indexer into two sets: a set restricted to terms in the Inspec thesaurus, and an uncontrolled set. Keywords were assigned from the full documents, and not simply the abstracts. The dataset was divided into 3 sets: 1000 for training, 500 for validation, and 500 for a hold out test set.

The specific implementation of the algorithm performs the following steps:

1. Sentence boundaries are detected and each sentence is separated.

2. Each sentence is part-of-speech (POS) tagged using the Stanford POS tagger (using tags from the Penn Treebank (Marcus et al., 1993)).

3. Vertices are chosen from the text based upon their POS tag (currently only nouns and adjectives).

4. Edges are drawn between vertices that fall within a co-occurrence window of size n:
   - Edges can be bidirectional, forward directional, or backward directional;
   - If edge weighting is used then the frequency count of that co-occurrence relation is used as the weight value.

5. The page rank algorithm is then run upon the constructed graph using initial values of 1 for each vertex until convergence to within a threshold occurs:
   - A threshold value of 0:0001 is used;
   - A damping factor d = 0:85 is used.

6. Vertices are sorted by their page rank scores in descending order and those k tokens are chosen as keywords:
   - Each token is expanded into a set of keyphrases by searching for each occurrence of the token in the original text, and for each occurrence collecting all adjacent words that are eligible tokens; and concatenating them into a phrase;
   - A keyphrase count value of k = floor($\frac{Total_Tokens}{3}$) is used, where Total_Tokens is the total number of tokens in each document.

3.4.1 Text as a Graph

The main steps of Mihalcea and Tarau research system is to construct a graph from the text, then apply the page rank algorithm on the constructed graph. Figure 3.2 shows

a document which is taken as an example from their dataset. The document name is (300.ABSTR) from Hulth (2003) dataset. First, every token in the document is tokenized and tagged as shown in Figure 3.3.

Abstract

The plot thins: thin-client computer systems and academic libraries.

The few libraries that have tried thin client architectures have noted a number of compelling reasons to do so. For starters, thin client devices are far less expensive than most PCs. More importantly, thin client computing devices are believed to be far less expensive to manage and support than traditional PC.

Manual Keywords and Keyphrases

Academic libraries

Thin-client

Computer systems

Figure 3.2: Example of an English Abstract, Title and its Actual Keywords and Keyphrases for the Document (300.ABSTR) in Hulth Dataset

The_DT plot_NN thins_VBZ :_: thin-client_JJ computer_NN systems_NNS and_CC academic_JJ libraries_NNS ._.

The_DT few_JJ libraries_NNS that_WDT have_VBP tried_VBN thin_JJ client_NN architectures_NNS have_VBP noted_VBN a_DT number_NN of_IN compelling_JJ reasons_NNS to_TO do_VB so_RB ._.

For_IN starters_NNS ,_, thin_JJ client_NN devices_NNS are_VBP far_RB less_RBR expensive_JJ than_IN most_JJS PCs_NNS ._.

More_RBR importantly_RB ,_, thin_JJ client_NN computing_NN devices_NNS are_VBP believed_VBN to_TO be_VB far_RB less_RBR expensive_JJ to_TO manage_VB and_CC support_VB than_IN traditional_JJ PCs_NNS ._.

Figure 3.3: A Tagged English Abstract for the Document (300.ABSTR) in Hulth Dataset

Then, nouns of types (NN, NNS, NNP and NNPS) and adjectives of types (JJ, JJR and JJS) as shown in Table 3.3 (Marcus et al., 1993) are selected to be the vertices to construct a graph of the above document. Co-occurrence relation is used to connect between the selected verrtices, controlled by the distance between word occurrences.

All lexical units that pass the above syntactic filter are added to the graph, and an edge is added between those lexical units that co-occur within a window of N words, where N can be set anywhere from 2 to 10 words (Mihalcea & Tarau, 2004). Figure 3.4 shows that the tokens shaded in yellow are selected as vertices from the document (300.ABSTR) in Hulth dataset to construct a graph after filtering them. Where the first part represents the token (word) and the second part represents its part of speech tag, note that they are seperated by underscore (_).

Figure 3.4 shows the shaded tokens in yellow that are selected as vertices from the document (300.ABSTR) in Hulth dataset to construct a graph after the above syntactic filter.

The_DT plot_NN thins_VBZ :_: thin-client_JJ computer_NN systems_NNS and_CC academic_JJ libraries_NNS ._.

The_DT few_JJ libraries_NNS that_WDT have_VBP tried_VBN thin_JJ client_NN architectures_NNS have_VBP noted_VBN a_DT number_NN of_IN compelling_JJ reasons_NNS to_TO do_VB so_RB ._.

For_IN starters_NNS ,_, thin_JJ client_NN devices_NNS are_VBP far_RB less_RBR expensive_JJ than_IN most_JJS PCs_NNS ._.

More_RBR importantly_RB ,_, thin_JJ client_NN computing_NN devices_NNS are_VBP believed_VBN to_TO be_VB far_RB less_RBR expensive_JJ to_TO manage_VB and_CC support_VB than_IN traditional_JJ PCs_NNS ._.

Figure 3.4: The Selected Vertices from Document (300.ABSTR) in Hulth Dataset

Table 3.3: Penn Treebank Tagset

POS Tag	Description	Example
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	there is
IN	prepositionsubordinating conjunction	in, of, like
JJ	adjective	green
JJR	adjective, comparative	greener
JJS	adjective, superlative	greenest
LS	list marker	1
MD	modal	could, will
NN	noun, singular or mass	table
NNS	noun plural	tables
NNP	proper noun, singular	John
NNPS	proper noun, plural	Vikings
PDT	predeterminer	both the boys
POS	possessive ending	friend's
PRP	personal pronoun	I, he, it
PRP$	possessive pronoun	my, his
RB	adverb	however, usually, naturally, here, good
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	particle	give up
TO	to	to go, to him
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, gerund/present participle	taking
VBN	verb, past participle	taken
VBP	verb, sing. present, non-3d	take
VBZ	verb, 3rd person sing. present	takes
WDT	wh-determiner	which
WP	wh-pronoun	who, what
WP$	possessive wh-pronoun	whose
WRB	wh-abverb	where, when

In general, edges in a graph may be directed or bidirectional. A bidirectional graph is graph that its vertices or nodes are connected together, where all the edges are bidirec-

tional, where each edge has two directions. In contrast, a graph where the edges point in a direction is called a directed graph (Insight, 2013). Directed graph may be forward or backward. In forward graphs, vertices are connected together based on a certain relation, where each vertex points to its successors. But in Backward graphs, each vertex points to its ancestors also according to a specific relation.

3.4.1.1 Bidirectional Graphs

Edges in bidirectional graphs are bidirectional, where each edge has two directions. Every word in the constructed graph point to its ancestors and successors according to the specified window size. Figure 3.5 shows the bidirectional graph built for the English document (300.ABSTR) in Hulth dataset with a window size=2.

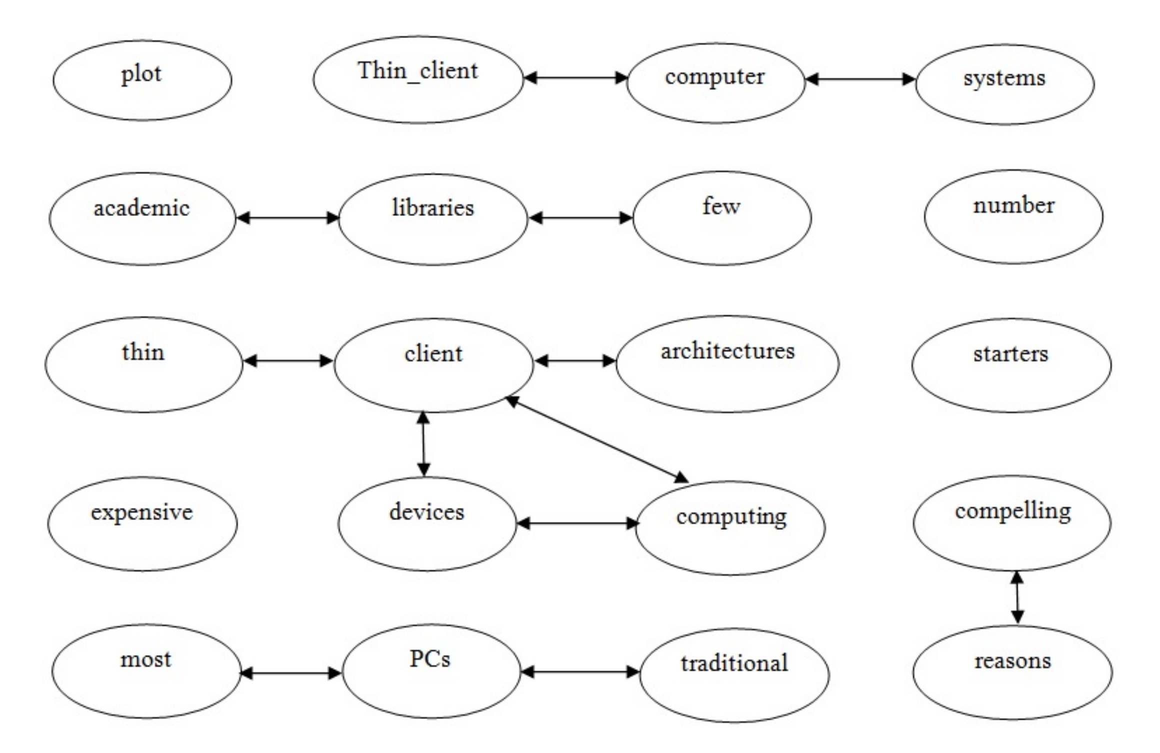
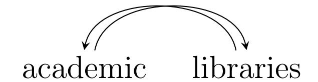


Figure 3.5: Bidirectional Graph for Document (300.ABSTR) in Hulth Dataset with Window Size=2

When the used window is set to 2, the shaded tokens (vertices of the constructed graph) by yellow in Figure 3.4 are tested to find if there existed vertices within a window size=2 in the original document. Then there will be a bidirectional relation between these vertices according to their locations in the original document. For example the words (academic) and (libraries) are both adjacent and selected as vertices in the constructed graph, so there will be a bidirectional relation between them as follows:

- (academic) will point to (libraries)
- (libraries) will point to (academic)

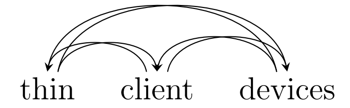
As shown below:



If the window size is set to 3, the shaded tokens (vertices of the constructed graph) by yellow in Figure 3.4 are tested again to find if there existed vertices within a window size=3 in the original document. Then there will be a bidirectional relation between these vertices according to their locations in the original document. For example the words (thin), (client) and (devices) are adjacents and selected as vertices in the constructed graph, so there will be a bidirectional relation between them.

- (thin) will point to (client)
- (client) will point to (thin)
- (thin) will point to (devices)
- (devices) will point to (thin)
- (client) will point to (devices)
- (devices) will point to (client)

As shown below:



3.4.1.2 Forward Graphs

Edges in a forward graph are directional. Every vertex points to its successors according to the used window size and its position in the text. Figure 3.6 shows the forward graph built for the English document (300.ABSTR) in Hulth dataset with a window size=2.

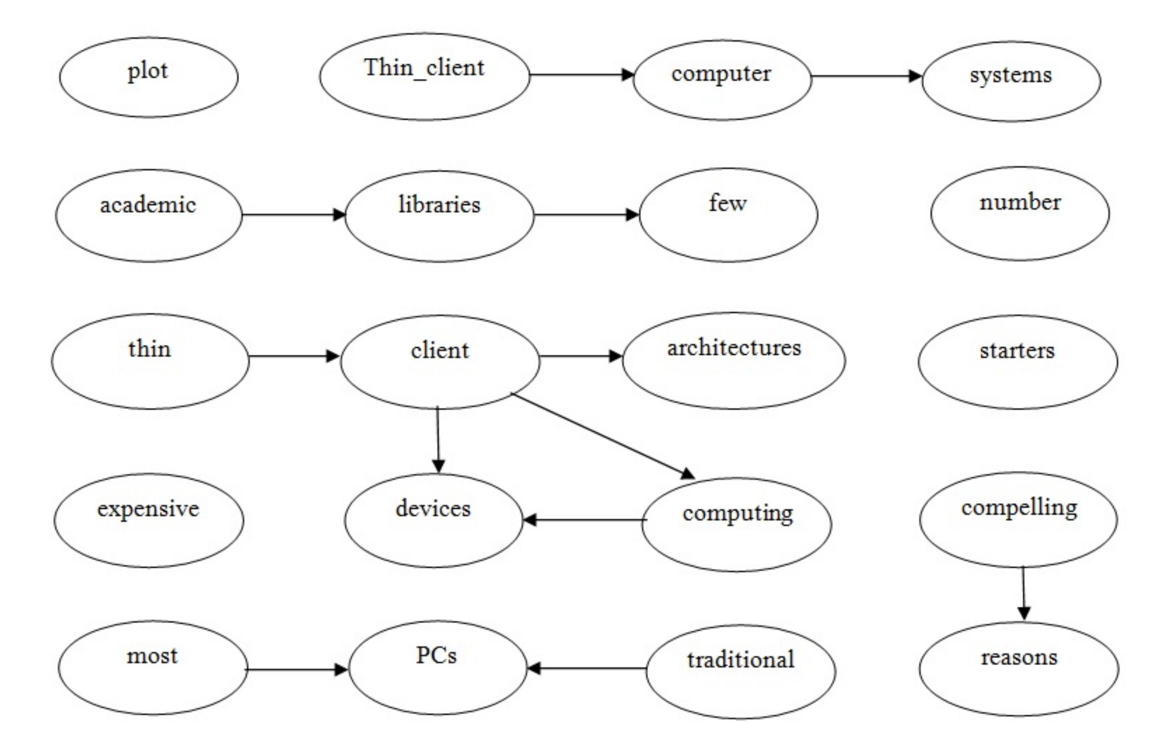
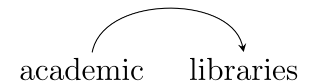


Figure 3.6: Forward Graph for the Document (300.ABSTR) in Hulth Dataset with Window Size=2

If the window size is set to 2, the shaded tokens (vertices of the constructed graph) by yellow in Figure 3.4 are tested to find if there existed vertices within a window size=2 in the original document. Then there will be a forward relation between these vertices according to their locations in the original document. For example the words (academic) and (libraries) are both adjacent and selected as vertices in the constructed graph, so there will be a forward relation between them as follows:

- (academic) will point to (libraries)

As shown below:

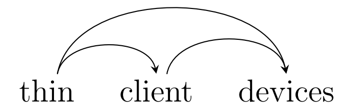


If the used window is set to 3, the shaded tokens (vertices of the constructed graph) by yellow in Figure 3.4 are tested again to find if there existed vertices within a window size=3 in the original document. Then there will be a forward relation between these vertices according to their locations in the original document. For example the words (thin), (client) and (devices) are adjacents and selected as vertices in the constructed graph, so there will be a forward relation between them.

- (thin) will point to (client)

- (thin) will point to (devices)
- (client) will point to (devices)

As shown below:



3.4.1.3 Backward Graphs

Contrary to what stated in forward graph, in backward graph, every vertex is pointing to its ancestors also based on the used window size and its position in the text. Figure 3.7 shows a backward graph for the English document (300.ABSTR) in Hulth dataset with window size=2.

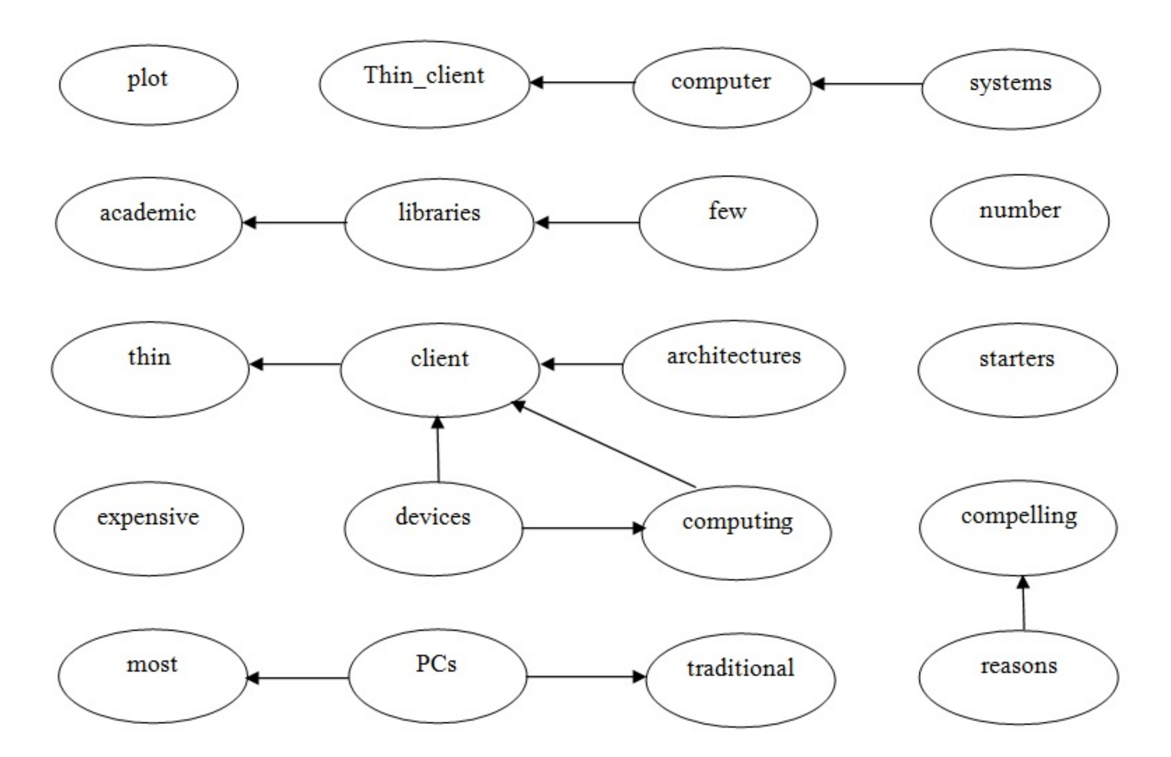
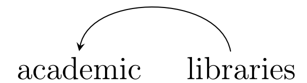


Figure 3.7: Backward Graph for the Document (300.ABSTR) in Hulth Dataset with Window Size=2

If the window size is set to 2, the shaded tokens (vertices of the constructed graph) by yellow in Figure 3.4 are tested to find if there existed vertices within a window size=2 in the original document. Then there will be a backward relation between these vertices according to their locations in the original document. For example the words (academic) and (libraries) are both adjacent and selected as vertices in the constructed graph, so there will be a backward relation between them as follows:

- (libraries) will point to (academic)

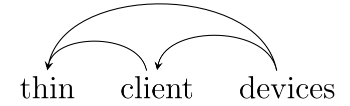
As shown below:



If the used window is set to 3, the shaded tokens (vertices of the constructed graph) by yellow in Figure 3.4 are tested again to find if there existed vertices within a window size=3 in the original document. Then there will be a backward relation between these vertices according to their locations in the original document. For example the words (thin), (client) and (devices) are adjacents and selected as vertices in the constructed graph, so there will be a backward relation between them.

- (client) will point to (thin)
- (devices) will point to (thin)
- (devices) will point to (client)

As shown below:



3.4.2 Applying Page Rank Algorithm

All lexical units that pass the syntactic filter are added to the graph, and edges are drawn between those lexical units that co-occur within a window of words. After the graph is constructed (Bidirectional, Forward or Backward). The score associated with each vertex is set to an initial value of 1. And the page rank Equation 3.1 described in Section 3.3 is run on the graph vertices for several iterations until it converges usually for 20-30 iterations, at a threshold of 0.0001 (Mihalcea & Tarau, 2004). Convergence is achieved when the error rate for any vertex in the graph falls below that threshold. Where this error rate is approximated with the difference between two successive scores for graph vertices.

Once a final score is obtained for each vertex in the graph in each document, vertices are sorted in descending order according to their page rank scores. The top 30% of the total tokens existed in each document are retained for post-processing steps for keywords and keyphrases extraction in a page rank list.

The basic idea implemented by a graph-based ranking model is that of voting. When

one vertex points to another one, it is basically creating a vote for that other vertex. The higher the number of votes that are created for a vertex, the higher the importance of the vertex. Hence, the score associated with a vertex is determined based on the votes that are created for it, and the score of the vertices creating these votes (Mihalcea & Tarau, 2004).

3.4.3 Post-Processing Steps to Extract Keywords and Keyphrases

After applying the above syntactic filters that select only lexical units of a certain part of speech, assigning a page rank score for each vertex in the constructed graph, arranging vertices in descending order according to their page rank scores and choosing the top 30% of the total number of words in the document as keywords. The post-processing stage to extract keywords and keyphrases can be detailed as follows:

1. Scan the original document token by token.

2. Add all single tokens found in the text and returned by the page rank algorithm to a candidate list of keyphrases with the corresponding page rank score.

3. Add all successive tokens found in the original text as a keyphrase to the keyphrases list with the corresponding page rank score which is equal to the summation of tokens page rank scores constituting that keyphrase.

4. Sort all the returned keywords and keyphrases based on their page rank scores.

To summarize the above steps, all tokens in the original document are scanned. If there existed two or more successive keywords then a keyphrase will be compoosed of these successive keywords, else a keyword of a single word is selected.

3.4.4 Weighted Graphs

Vertices are connected through links (edges). These links may be strong or poor depending on several factors. For example, words may occur more than once in the text, or there exist some words have greater pagerank scores pointing to other words. These links between these words are more strong, which means a high pagerank score may be obtained.

The strength of the connection between two vertices V_i and V_j is expressed as a weight W_{ij} for such relations. This weight can be added to the link between two vertices to high the pagerank score (Mihalcea & Tarau, 2004). So a new formula of pagerank equation

that takes the algorithm is formed to compute ranks by taking into account the weight as shown in equation 3.3.

$$WS(V_i) = (1 - d) + d * \sum_{V_j \in In(V_i)} \frac{W_{ij}}{\sum_{V_k \in Out(V_j)} W_{jk}} WS(V_j) \tag{3.3}$$

To illustrate what came in Equation 3.3, suppose the links between two words are shown in Figure 3.8. The relation in A can be considered stronger than the relation in B, Edge weight can be taken into account as a measure to compute the pagerank score associated with each vertex in the graph.

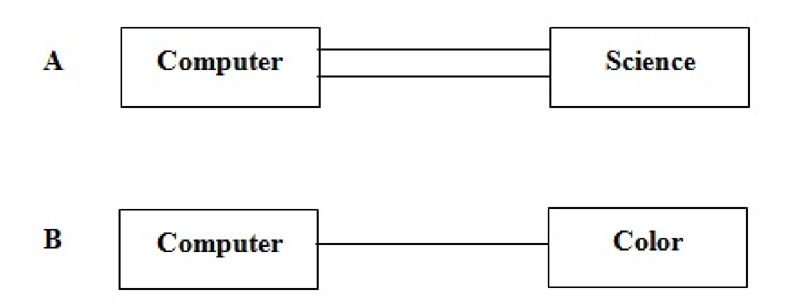


Figure 3.8: Example on Weighted Page Rank Scores

3.5 Evaluation Methods

The results are evaluated using precision, recall, and F-measure. As a note, the maximum recall that can be achieved on this research is less than 100%, since many actual built keywords and keyphrases are not existed in the documents (abstracts and its titles). Precision and recall are the basic measures used in evaluating search strategies (Mihalcea & Tarau, 2004).

Recall as in Equation 3.5 is the ratio of the number of relevant keywords or keyphrases retrieved to the total number of relevant keywords in the file. It is usually expressed as a percentage (Creighton, 2013).

$$Recall = R/N \tag{3.4}$$

Where R is the number of relevant keywords and keyphrases retrieved, and N is the number of relevant keywords in the collection.

Precision as in Equation 3.6 is the ratio of the number of relevant keywords retrieved to the total number of irrelevant and relevant keywords retrieved. It is also usually expressed as a percentage (Creighton, 2013). Recall and precision are inversely related. It means

when Recall has increased the precision will decrease, and when Recall has decreased the precision will increase. High recall means that an algorithm returned most of the relevant keywords, while high precision means that an algorithm returned substantially more relevant keywords than irrelevant.

$$Precision = R/D \tag{3.5}$$

R is the number of relevant keywords and keyphrases retrieved and D is the number of keywords and keyphrases retrieved.

A measure that combines precision and recall is called the F-measure or balanced F-score. It can be calculated as follows (Manning & Schutze, 2000):

$$F = 2 * \left(\frac{Recall * Precision}{Recall + Precision}\right) \tag{3.6}$$

This measure can be considered as a measure of a test's accuracy. Since it considers both the precision and the recall of the test. Recall, Precision and F-measure are often used in information retrieval field for measuring search, document classification, and query classification performance.

Chapter 4

Literature Review

4.1 Introduction

This chapter details the history of keywords and keyphrases extraction systems and the previous works done in this field. Breif overview is presented. All approaches in keywords extraction are viewed. Keywords and keyphrases extraction systems on English, Arabic and other language are mentioned.

4.2 Overview

Manual keywords extraction process is very slow, exhausted, expensive and prone for mistakes. Therefore, many automatic algorithms and systems for keywords extraction have been proposed. Existing methods are divided into four categories Oelze (2009):

- Simple statistics approaches
- Linguistics approaches
- Machine learning approaches
- Mixed approaches

4.3 Simple Statistics Approaches

These methods are simple, they focus on non-linguistic features of the text such as term frequency, inverse document frequency, and the keyword position. Besides, they have limited requirements and do not need the training data. Words statistics can be used to identify the keywords in the document. The benefits of purely statistical methods are their ease of use and the fact that they do generally produce good results (Oelze, 2009).

Weighting a term by occurrence dates back to the 1950s in the study by Luhn (Luhn, 1957).

Salton and his parteners (Salton et al., 1975) have proposed the TF*IDF method. They have used techniques in their work that were based on word frequency characteristics which applied largely in an ad hoc manner. They concluded that exhibiting high occurence frequencies in individual documents were often useful for high recall performance, whereas terms with low frequency in the whole collection were useful for high precision.

Cohen used N-Gram statistical information to automatically index the document by applying a statistical method of highlighting index terms from text (Cohen, 1995).

Matsuo and Ishizuka have presented a keywords extraction algorithm that is applied to a single document without using a corpus, where frequent terms are extracted first. Then co-occurrences of a term and frequent terms are counted. They concluded that if a term appears frequently with a particular subset of terms, the term is likely to have important meaning and can be considered as a keyword. (Matsuo & Ishizuka, 2004).

Jiao and his parteners presented a Chinese text encoding method based on Chinese word, and established a Chinese document format which dealt with the automatic segmentation issue. In their work, N-gram and word co-occurrence statistical analysis were combined to carry out Chinese keywords extraction experiment. Candidate keywords were extracted with bi-gram model. Then, a set of co-occurrences between every word in bi-grams and frequent words was generated. According to the analysis result, keywords were chosen from bi-grams. This algorithm was applied to a single document without using a corpus, and experimental results were satisfying (Jiao et al., 2007).

A text mining technique for automatically extracting association rules was proposed by Mahgoub and his friends on a collections of textual documents . They called the technique Extracting Association Rules from Text (EART). they have used n keyword features to discover association rules amongst keywords labeling the documents. In their work, the EART system ignored the order in which the words occur, but instead focusing on the words and their statistical distributions in documents. They integrated XML technology with Information Retrieval scheme (TF-IDF) and use Data Mining technique for association rules discovery (Mahgoub et al., 2008).

Wartena, Brussee and Slakhorst have used relations between words. The alternative relevance measures are computed by defining co-occurrence distributions for words and comparing these distributions with the document and the corpus distribution. For two cor-

pora of abstracts with manually assigned keywords, they compared manually extracted keywords with different automatically extracted ones. Their results showed that using word co-occurrence information can improve precision and recall over tf.idf (Wartena et al., 2010).

4.4 Linguistics Approaches

These approaches use the linguistic features of the words, sentences and document. Methods which focus on linguistic features such as part-of-speech, syntactic structure and semantic qualities. All of these features tend to add value, functioning sometimes as filters for bad keywords (Oelze, 2009).

Hulth (Hulth, 2003a) used different methods of incorporating linguistics into keywords extraction. Besides, terms are selected as keywords using three features: Document frequency (TF), collection frequency (IDF), relative position of its first occurrence in a document and the term's part of speech tag. she added linguistic knowledge to the representation (such as syntactic features), rather than relying only on statistics (such as term frequency and n-grams), a better result is obtained as measured by keywords previously assigned by professional indexers.

Lexical resources such as WordNet and the EDR electronic dictionary have been used in several natural language processing tasks. WordNet has been used far more often than the EDR. Plas and his friends (Plas et al., 2004) have used both resources on the same task. The task was automatic assignment of keywords to multi-party dialogue episodes. They showed that the use of lexical resources in such a task improved the performances than the use of a purely statistically based method.

Xinghua and Wu propsed (Hu & Wu, 2006) a position weight (PW) algorithm for keywords extraction that utilized linguistic features to represent the importance of the word position in a document. Topical terms and their previous-term and next-term co-occurrence collections are extracted. They measured the degree of correlation between a topical term and its co-occurrence terms by using three methods: Term frequency inverse term frequency (TFITF), position weight inverse position weight (PWIPW), and CHI-square (Chi2). The co-occurrence terms that have the highest degree of correlation and exceeded a co-occurrence frequency threshold were combined together with the original topical term to form a final keyword.

In Ercan and Cicekli paper (Ercan & Cicekli, 2007) have used a lexical chain that holded a set of semantically related words of a text and it could be said that a lexical chain

represented the semantic content of a portion of the text. Although lexical chains have been extensively used in text summarization, their usage for keywords extraction problem has not been fully investigated. In their work, a keywords extraction technique that used lexical chains was described, and encouraging results were obtained.

A keywords extraction method named "Tag-Based Keyword Extraction" was proposed by Zhao and his friends (Zhao et al., 2010) to extract keywords with the help of tags. Their experiment results showed that their method can be compared to the existing keywords extraction methods remarkably. They utilized linguistic features and/or statistical features of texts to accomplish keywords extraction with other valuable information.

For Arabic language Hammadi and Aziz had proposed a method to recognize Grammatical Relations (GRs), as the rule-based approach had been successfully used in developing many natural language processing systems. (GR) can be defined as a linguistic relation established by grammar, where linguistic relation is an association among the linguistic forms or constituents. The proposed technique enhanced the basic representations of Arabic language such as: Noun Phrase (NP), Verb Phrase (VP), Preposition Phrase (PP) and Adjective Phrase (AP). They had implemented and evaluated the Rule-Based approach that handles chunking and GRs of Arabic sentences (Hammadi & Aziz, 2012).

4.5 Machine Learning Approaches

Keywords extraction can be considered as supervised learning. The machine learning mechanism works as follows. First a set of training documents is provided to the system, each of which has a range of human-chosen keywords as well. Then the gained knowledge is applied on new test documents to find keywords (Oelze, 2009).

A method for keywords extraction of radio news was proposed by Suzuki and his friends (Suzuki et al., 1998). There were two procedures in their work: Term weighting and keywords extraction. In terms of of term weighting, a feature vector of each domain was calculated using an encyclopedia and newspaper articles. And in terms of of keywords extraction, keywords were extracted using feature vectors and result of domain identification. The results of experiments demonstrated the applicability of the method.

Keyphrases provide semantic metadata that summarize and characterize documents. Witten and his parteners (Witten et al., 1999) have described Kea, an algorithm for automatically extracting keyphrases from text. Kea identified candidate keyphrases using lexical

methods, calculated feature values for each candidate, and used a machine-learning algorithm to predict which candidates are good keyphrases. They used a large test corpus to evaluate Kea's effectiveness in terms of how many author assigned keyphrases are correctly identified.

Mihalcea and Tarau (Mihalcea & Tarau, 2004) have introduced TextRank – a graph-based ranking model for text processing, and showed how this model can be successfully used in natural language applications. They proposed two innovative unsupervised methods for keywords and sentences extraction, and showed that the results obtained can be compared favorably with previously published results.

A study by Krishnan and his parteners (Krishnan et al., 2010) was introduced at designing a support vector machine (SVM)-based classifier for breast cancer detection with higher degree of accuracy. It introduced a best possible training scheme of the features extracted from the mammogram, by first selecting the kernel function and then choosing a suitable training-test partition. A comparative study has been performed in respect to diagnostic measures, confusion matrix, sensitivity and specificity. They have considered two data sets from UCI machine learning database having nine and ten dimensional feature spaces for classification.

Tiwari and his parteners (Tiwari et al., 2010) have described a framework to extract precise information about coexpression relationship among genes, from published literature using a supervised machine learning approach. They used a graphical model, Dynamic Conditional Random Fields (DCRFs), for training their classifier. Their approach was based on semantic analysis of text to classify the predicates describing coexpression relationship rather than detecting the presence of keywords.

4.6 Mixed Approaches

Keywords extraction can be mainly applied by combining the methods mentioned above or use some heuristic knowledge in the task of keywords extraction, such as the position, length, layout feature of the words, html tags around of the words, etc (Oelze, 2009).

Keyphrases are an important means of document summarization, clustering, and topic search. Only a small minority of documents have author-assigned keyphrases, and manually assigning keyphrases to existing documents is very laborious. Therefore it is highly desirable to automate the keyphrase extraction process. Frank and his parteners (Frank et al., 1999) has shown that a simple procedure for keyphrase extraction based on the naive bayes scheme performed comparably to the state of the art. It explained how this

procedure's performance can be helpful by automatically tailoring the extraction process to the particular document collection on a large collection of technical reports in computer science. They have shown that the quality of the extracted keyphrases improved significantly when domain-specific information is exploited.

A collecction of Arabic abstracts with their titles is used by treating the words within the text as web pages. This method is called keywords extarction using text rank model. Keywords extraction from abstracts is more widely informative than from full texts, since many documents on the Internet are not available as full-texts, but only as abstracts. TextRank, as outlined in Mihalcea and Tarau (Mihalcea & Tarau, 2004) as mentioned before will be implemented on the Arabic collection.

Zhang and his friends (Zhang et al., 2006) have proposed in addition of using "global context information" a "local context information" for extracting keywords from documents. Methods for performing the tasks on the basis of support vector machines have also been proposed in this paper. Features in the model have been defined. The proposed method has been applied to document classification, a typical text mining processing. Experimental results showed that the accuracy of document classification can be significantly improved by using the keywords extraction method.

Dias and Malheiros have adapted an algorithm for automatic extraction of keywords for the Portuguese Language. In their work they focused on the extraction of keywords for theses on several fields of knowledge. The KEA algorithm was used, together with a stemming technique specific to Portuguese and a manually created list of stopwords. Their results that were obtained were good enough for practical use.

Al-Shalabi and his friends (AL-Shalabi et al., 2006) have performed keywords extraction as a stage of their work in text categorization for Arabic text using key Nearest Neighbor (KNN) algorithm. In their project they have implemented the key Nearest Neighbor (KNN) algorithm, which is known to be one of top performing classifiers applied for the English text along with the Support Vector Machines (SVMs) algorithm. However the nature of Arabic text is different than that of the English text and the preprocessing of the Arabic text is different and more challenging. And for the problem of keywords extraction and reduction they have implemented a method to extract keywords based on the Document Frequency threshold (DF) method after applying a light stemming process on tokenized words.

In El-Beltagy and Rafea paper (El-Beltagy & Rafea, 2009), they presented the KP-Miner system and showed through experimentation and comparison with widely used

systems that using KP-Miner system is effective and efficient in extracting keyphrases from both English and Arabic documents of varied length. It also has the advantage of being configurable as the rules and heuristics adopted by the system are related to the general nature of documents and keyphrases.

In El-Shishtawy and Al-sammak paper (El-shishtawy & Al-sammak, 2009), a supervised learning technique for extracting keyphrases of Arabic documents was presented. Linguistic feature was used to enhance its efficiency instead of relying only on statistical information such as term frequency and distance. They have an annotated Arabic corpus to extract the required lexical features of the document words. They also applied a syntactic rules based on part of speech tags and allowed word sequences to extract the candidate keyphrases. They used the abstract form of Arabic words instead of its stem form to represent the candidate terms because it hided most of the inflections found in Arabic words. Their work introduced features of keyphrases based on linguistic knowledge, to capture titles and subtitles of a document. A simple ANOVA test was used in their work to evaluate the validity of selected features. Then, the learning model is built using the LDA - Linear Discriminant Analysis - and training documents.

A technique to produce a summary of an original text was investigated in Al-Hashemi paper (Al-Hashemi, 2010). His model consisted of four stages. The preprocess stages converted the unstructured text into structured. In the first stage, their system removed the stop words and assigning the POS (tag) for each word in the text and store the result in a table. The second stage was to extract the important keyphrases in the text by implementing an algorithm through ranking the candidate words . The system used the extracted keywords/keyphrases to select the important sentence . They have used similarity measurements and features such as the existence of the keywords/keyphrase in it, the relation between the sentence and the title. The system used Term frequency, inverse document frequency and words existence in the document title and font type to distinguish keywords . The Third stage of their proposed system was to extract the sentences with the highest rank. The Forth stage was the filtering stage which reduced the amount of the candidate sentences in the summary in order to produce a qualitative summary using KFIDF measurement.

The identification of collocations is very important part in natural language processing applications. Because of the complexities of Arabic, the collocations faced some variations such as, morphological, graphical, syntactic variation that constitutes the difficulties of identifying the collocation. Saif and Aziz in (Saif & Aziz, 2011) used the hybrid method for extracting the collocations from Arabic corpus. Their method was based on linguistic information and association measures. Their system extracted the bi-gram candidates

of Arabic collocation from corpus and evaluated the association measures by using the n-best evaluation method. The experimental results showed that the log-likelihood ratio is the best association measure that achieved highest precision.

An automatic keywords extraction system has proposed for Punjabi language text by Gupta and Lehal in (Gupta & Lehal, 2011). It included various phases like removing stop words, identification of Punjabi nouns and noun stemming, calculation of Term Frequency and Inverse Sentence Frequency (TF-ISF). The extracted keywords were very much helpful in automatic indexing, text summarization, information retrieval, classification, clustering, topic detection, tracking and web searches.

El-Shishtawy and El-Ghannam (El-Shishtawy & El-Ghannam, 2012) described a computationally inexpensive and efficient generic summarization algorithm for Arabic texts. Their algorithm belongs to extractive summarization family. Important keyphrases of the document to be summarized are identified employing combinations of statistical and linguistic features. The sentence extraction algorithm exploited keyphrases as the primary attributes to rank a sentence. Their keyphrase extractor used in their work is based on the existing Arabic keyphrase extractor AKE (El-shishtawy & Al-sammak, 2009) that was proposed also by El-Shishtawy and Al-Sammak.

Chapter 5

Keywords Extraction Using Page Rank Algorithm for Arabic Text - The Research Methodology

5.1 Introduction

This chapter is concerned with the methodology and implementation of the keywords and keyphrases extraction system using page rank algorithm. It starts by explaining the proposed system algorithm architecture. The basic steps of the algorithm implementation using bidirectional, forward and backward graphs are clarified. Additional steps added to the basic system implementation such as stopwords removal and Part Of Speech (POS) tagging process are also clarified .All the rules that are used in the system are listed and discussed. Finally the features of the system are mentioned.

5.2 Overall Architecture

The main purpose of the keywords extraction system is to automatically identify a set of terms and phrases that best describes the document. Such keywords and keyphrases may constitute useful entries for building an automatic index for a document collection. And can be used to classify a text, or may serve as a concise summary for a given document.

The proposed system for keywords and keyphrases extraction system consists of major components. These components are depicted in Figure 5.1. The input to the system is an Arabic abstract document. The output is a set of keywords and keyphrases that describe the content of this document.

Additional steps may be depicted to the main architecture at words selection stage to

build vertices of the graph. Stopwords removal is added to improve results since they rarely appear in keywords and keyphrases sets. Also POS Tagging process can be useful to filter the document words and nominate particular group of words. As an explanation verbs and particles can be excluded and just nouns are nominated for the graph vertices. these additional steps can be useful to improve the results.

Experiments were gradually applied step by step to compare the results clearly. First main steps that are clarrified in the general architecture are applied for the three types of graph: Forward, backward and bidirectional. Then they are tested for different window sizes to decide which graph type and window size can lead to better results. After that the additional steps are added such as stopwords removal and tagging to see what changes can these processes affect the system.

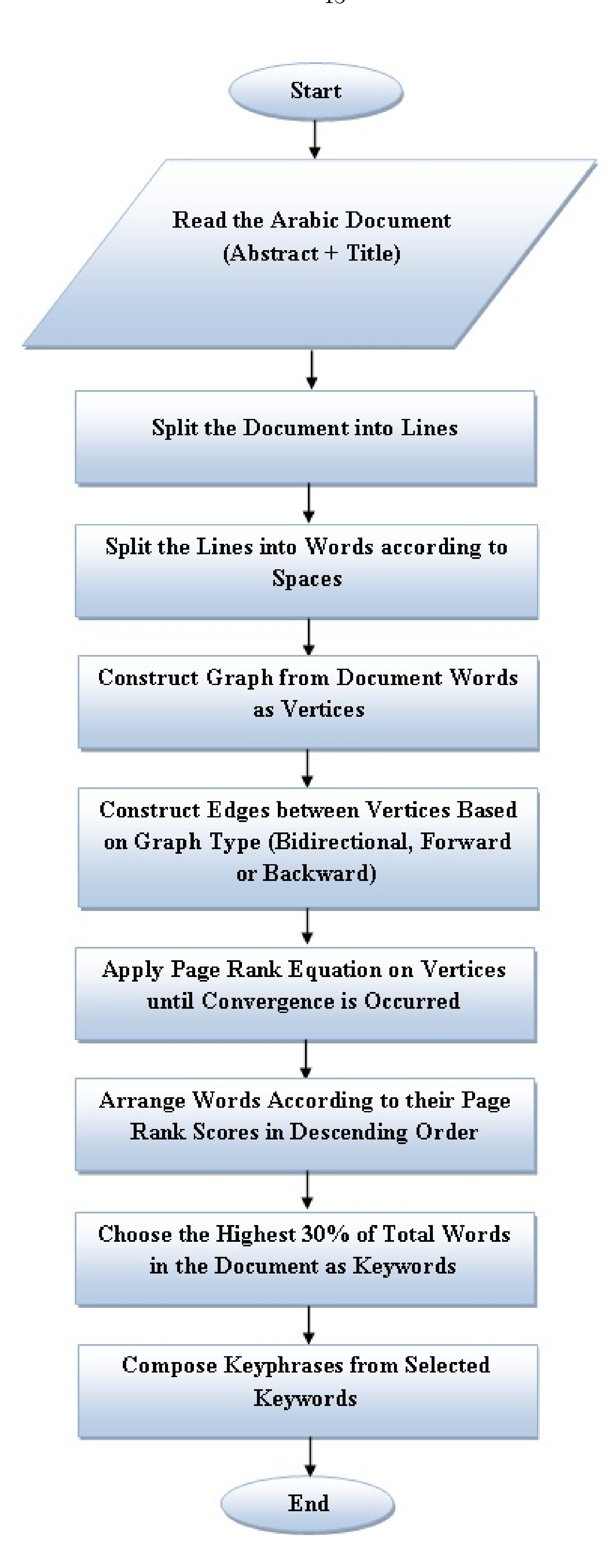


Figure 5.1: The General Structure of a Keywords and Keyphrases System using Page Rank Algorithm on Arabic Text

Finally according to several experiments, Keyphrases that is constituted of one, two and three words are just selected as it will be discussed in details with reasons and proofs in Chapter 5.

5.3 Keywords and Keyphrases Extraction System using Page Rank Algorithm on Arabic Text

This section details the implementation of keywords and keyphrases extraction system using page rank algorithm that is shown in Figure 5.1 on a collected Arabic dataset. This collected dataset consists of 150 Arabic articles from different disciplines. This dataset was divided into 2 sets: 100 documents for training, and 50 documents for a hold out test set which are being used in this research. Documents abstracts and their titles are being collected and printed in Notepad files with (.ABSTR) extensions. Their actual keywords have been printed in other Notepad files with the same names of the abstracts files but with different extensions (.UNCONTR).

Keywords and keyphrases extraction system using Google page rank algorithm (Text Rank Model) consists the following steps on each documnet (abstract and its title) in the used dataset:

1. Arabic Sentences boundaries are detected and each sentence is separated.
2. Arabic words in each sentence are tokenized and tagged manually into three main tags: Verbs(V), Nouns(N, N1 and NT) and Particles(P).
   All thanks and appreciation to Dr. Abd Al-Haleem Mohammad Al Ramahi, a doctor and consultant in the Arabic language and Islamic law, Where it was back to him in Arabic tagging process.
3. Arabic nouns are divided into three main types:
   - (N) refers to general nouns that are not attached to any type of particles
   - (N1) refers to nouns that are attached to particles such as pronouns or attributes that describe these nouns
   - (NT) refers to general nouns that are not attached to any type of particles and appeared in documents titles
4. Punctuation marks between words in each documnet are tokenized as punctuation marks that seperate sentences in files.
5. Vertices are chosen from the text based upon their manual part of speech tagging, only nouns with types (N and NT) are selected. According to the tarining set, user

tends to use nouns as keywords in Arabic language. In contrast to English, where user tends to use nouns and adjectives as keywords.

6. Edges are drawn between vertices that fall within a co-occurrence window of size n, edges can be forward, backward or bidirectional.

7. The page rank algorithm is applied on the constructed graph using initial values of (Lawrence et al., 1998) for each vertex until convergence to within a specified threshold value occurs.

8. Vertices are sorted by their page rank scores in descending order and a rate of the total number of tokens in each document is chosen as keywords. Each token is expanded into a set of keyphrases by searching for each occurrence of the token in the original text, and for each occurrence collecting all adjacent words that are eligible tokens and concatenating them into a phrase (Mihalcea & Tarau, 2004). In this research, the top 30% of the total number of tokens in each document will be taken as keywords according to their page rank scores. For example if the total number of words in a document were 100, then just the top 30 vertices will be taken as keywords. And if the total number of words in a document were 85, a ceil function is used to ignore the fractional part and round up to the next integer, then the top 26 verices will be chosen as keywords.

5.3.1 Text as a Graph for Arabic Text

The main steps of this research system is to construct a graph from the text, then apply the page rank algorithm on the constructed graph. Figure 5.2 shows an Arabic document which is taken as an example from the collected dataset. The document name is (42.AB-STR). First, every token in the document is tokenized and tagged as shown in Figure 5.3.

(Title)

العينة في عقود التبرعات : الهبة نموذجا

(Abstract)

تناولت هذه المقالة مسألة انعقاد عقود التبرعات ، هل يكون بمجرد الإيجاب و القبول ، أم لا بد من حدوث القبض لحصوله ؟ أو ما يعرف عند القانونيين بمسألة عينية العقود ، و دراستها دراسة مقارنة بعرض الأقوال في الفقه الإسلامي و أدلتها الخلوص إلى الراجح منها .

(Manual Keywords and keyphrases)

العينية

عقود التبرعت

الهبة

Figure 5.2: An Example of the Arabic Abstract Document (42.ABSTR)

العينية_NT في_P عقود_NT التبرعات_NT :_: الهبة_NT نموذجا_N1

تناولت_V هذه_P المقالة_N مسألة_N انعقاد_N عقود_N التبرعات_N ،_، هل_P
يكون_V بمجرد_N1 الإيجاب_N و_P القبول_N ،_، أم_P لا_P بد_P من_P حدوث_N
القبض_N لحصوله_N1 ؟_؟ أو_P ما_P يعرف_V عند_P القانونيين_N بمسألة_N1
عينية_N العقود_N ،_، و_P دراستها_N1 دراسة_N مقارنة_N بعرض_N1 الأقوال_N
في_P الفقه_N الإسلامي_N و_P أدلتها_N1 الخلوص_N إلى_P الراجح_N منها_P ._.

Figure 5.3: Tokens and their Part-Of-Speech Tags for the Arabic Abstract Document (42.ABSTR)

Then, nouns of types (N and NT) are selected to be the vertices after stopwords removal to construct a graph of the above document. Co-occurrence relation is used to connect between the selected verrtices, controlled by the distance between word occurrences. All lexical units that pass the above syntactic filter are added to the graph, and an edge is added between those lexical units that co-occur within a window of N words,

where N can be set anywhere from 2 to 10 words. Figure 5.4 shows that the tokens shaded in yellow are selected as vertices from the document (42.ABSTR) to construct a graph after filtering them and just choosing the tokens of type N or NT. Where the first part represents the token (word) and the second part represents its part of speech tag, note that they are seperated by underscore (_).

العينية_NT في_P عقود_NT التبرعات_NT :_: الهبة_NT نموذجا_N1

تناولت_V هذه_P المقالة_N مسألة_N انعقاد_N عقود_N التبرعات_N ،_، هل_P
يكون_V بمجرد_N1 الإيجاب_N و_P القبول_N ،_، أم_P لا_P بد_P من_P حدوث_N
القبض_N لحصوله_N1 ؟_؟ أو_P ما_P يعرف_V عند_P القانونيين_N بمسألة_N1
عينية_N العقود_N ،_، و_P دراستها_N1 دراسة_N مقارنة_N بعرض_N1 الأقوال_N
في_P الفقه_N الإسلامي_N و_P أدلتها_N1 إلى_P الخلوص_N الراجح_N منها_P ._.

Figure 5.4: The Selected Verices from Document (42.ABSTR)

As in English text, edges in a graph may be directed or bidirectional. A bidirectional graph is graph that its vertices or nodes are connected together, where all the edges are bidirectional. In contrast, a graph where the edges point in a direction is called a directed graph. Directed graph may be forward or backward. In forward graphs, vertices are connected together based on a certain relation, where each vertex points to its successors. But in backward graphs, each vertex points to its ancestors also according to a specific relation.

5.3.2 Graph Types

All the graph types are implemented seperately on the training set. Then other tools will be added for each graph type gradually. Results are compared to determine the best ones depending on the highest average of F-Measure values. After choosing the best settings from these experiments, the selected settings will be applied on the test set and adopted as the final ones.

The first experiment was to implement the above steps of the algorithm in section 5.2 and construct a graph with bidirectional, forward and backward graphs on the collected dataset. To decide which graph type led to the best results based on F-measure value. Document (42.ABSTR) is taken from the collected dataset as an example to illustrate how graph with all its types is constructed.

5.3.2.1 Bidirectional Graphs

Each edge in bidirectional graph has two directions. Figure 5.5 shows the bidirectional graph built for the Arabic document (42.ABSTR) with a window size=2.

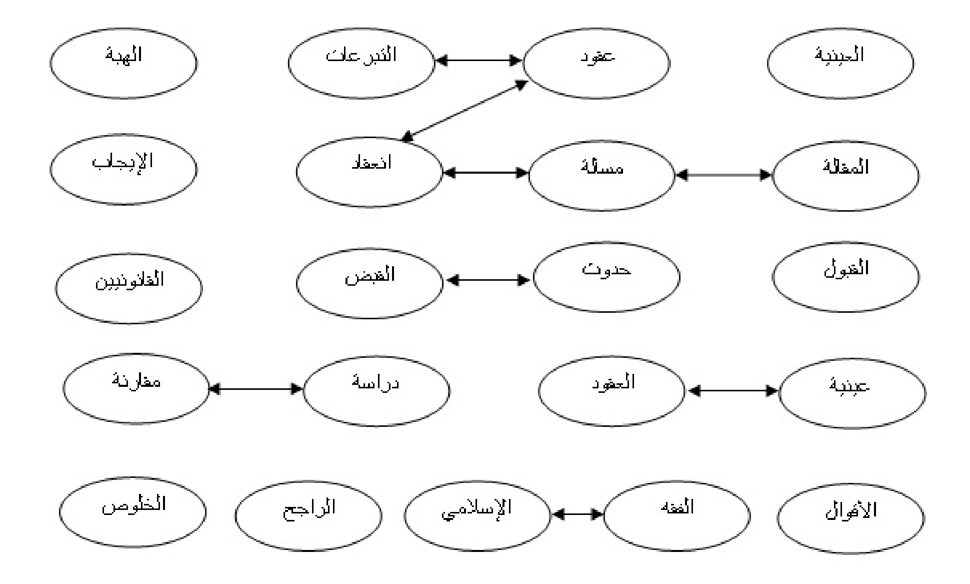


Figure 5.5: Bidirectional Graph for Document (42.ABSTR) with Window Size=2

When the window size is set to 2, the shaded tokens (vertices of the constructed graph) by yellow in Figure 5.4 are tested to find if there existed vertices within a window size=2 in the original document. Then there will be a bidirectional relation between these vertices according to their locations in the original document. For example the words (**عقود**) and (**التبرعات**) are both adjacent and selected as vertices in the constructed graph, so there will be a bidirectional relation between them as follows:

- (**عقود**) will point to (**التبرعات**)
- (**التبرعات**) will point to (**عقود**)

As shown below:

عقود التبرعات

If the window size has increased to 3, the result graph will be reconstructed as shown in Figure 5.6.

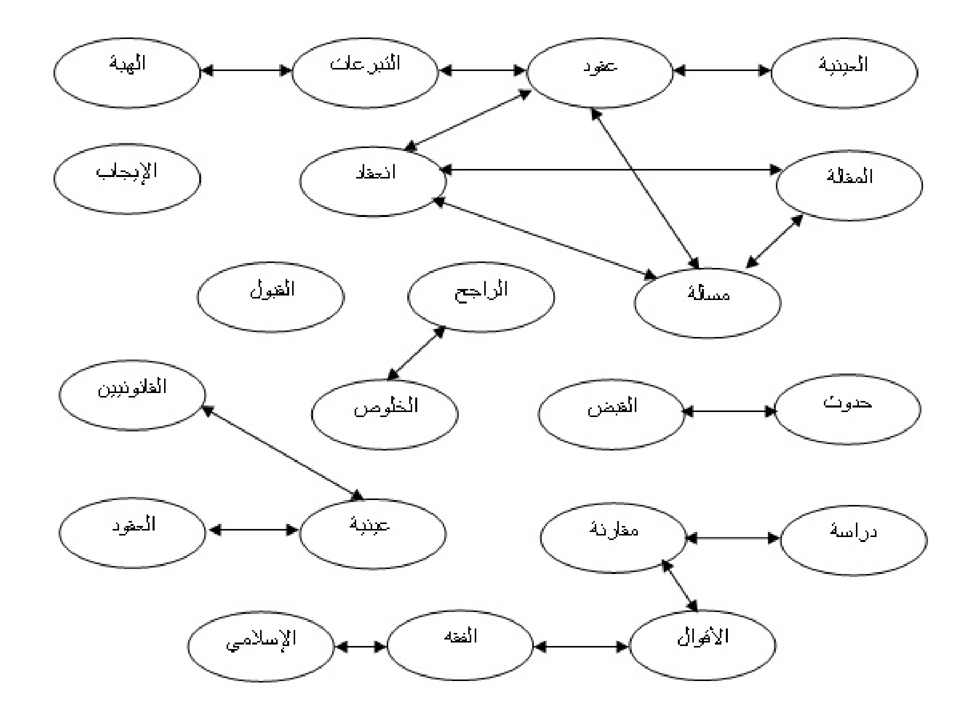
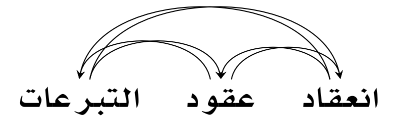


Figure 5.6: Bidirectional Graph for the Document (42.ABSTR) with Window Size=3

If the window size is set to 3, the shaded tokens (vertices of the constructed graph) by yellow in Figure 5.4 are tested again to find if there existed vertices within a window size=3 in the original document. Then there will be a bidirectional relation between these vertices according to their locations in the original document. For example the words (انعقاد), (عقود) and (التبرعات) are adjacents and selected as vertices in the constructed graph, so there will be a bidirectional relation between them.

- (انعقاد) will point to (عقود)
- (عقود) will point to (انعقاد)
- (انعقاد) will point to (التبرعات)
- (التبرعات) will point to (انعقاد)
- (عقود) will point to (التبرعات)
- (التبرعات) will point to (عقود)

As shown below:



5.3.2.2 Forward Graphs

Every vertex are pointing to its successors according to the used window size and its position in the text. Figure 5.7 shows the forward graph built for the Arabic document (42.ABSTR) with a window size=2.

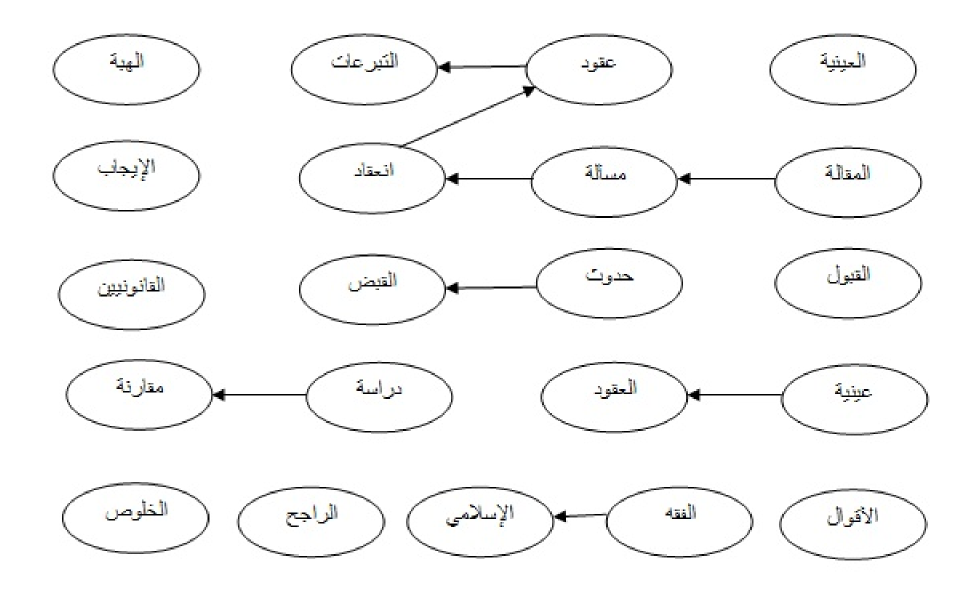


Figure 5.7: Forward Graph for the Document (42.ABSTR) with Window Size=2

If window size is set to 2, the shaded tokens (vertices of the constructed graph) by yellow in Figure 5.4 are tested to find if there existed vertices within a window size=2 in the original document. Then there will be a forward relation between these vertices according to their locations in the original document. For example the words (**عقود**) and (**التبرعات**) are both adjacent and selected as vertices in the constructed graph, so there will be a forward relation between them as follows:

- (**عقود**) will point to (**التبرعات**)

As shown below:



Figure 5.8 shows the same forward graph example, but with window size=3.

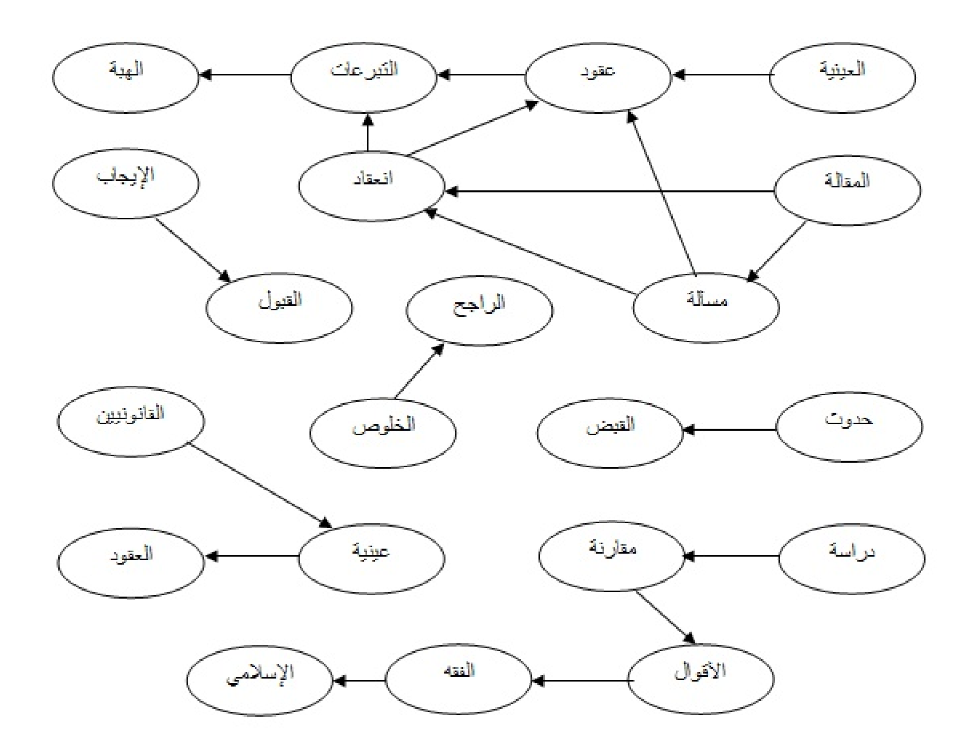
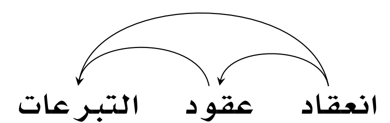


Figure 5.8: Forward Graph for the Document (42.ABSTR) with Window Size=3

And window size is set to 3, the shaded tokens (vertices of the constructed graph) by yellow in Figure 5.4 are tested again to find if there existed vertices within a window size=3 in the original document. Then there will be a forward relation between these vertices according to their locations in the original document. For example the words (انعقاد), (عقود) and (التبرعات) are adjacents and selected as vertices in the constructed graph, so there will be a forward relation between them.

- (انعقاد) will point to (عقود)
- (انعقاد) will point to (التبرعات)
- (عقود) will point to (التبرعات)

As shown below:



5.3.2.3 Backward Graphs

Contrary to what stated in forward graph, in backward graph, every vertex is pointing to its ancestors also based on the used window size and its position in the text. Figure 5.9 shows a backward graph for the Arabic document (42.ABSTR) with window size=2.

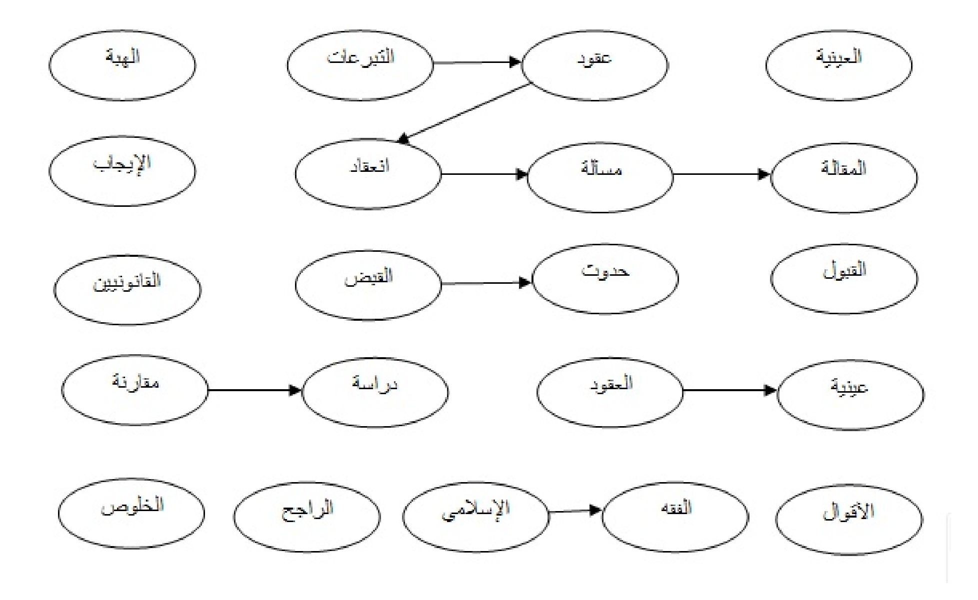


Figure 5.9: Backward Graph for the Document (42.ABSTR) with Window Size=2

If window size is set to 2, the shaded tokens (vertices of the constructed graph) by yellow in Figure 5.4 are tested to find if there existed vertices within a window size=2 in the original document. Then there will be a backward relation between these vertices according to their locations in the original document. For example the words (**عقود**) and (**التبرعات**) are both adjacent and selected as vertices in the constructed graph, so there will be a backward relation between them as follows:

- (**التبرعات**) will point to (**عقود**)

As shown below:

عقود التبرعات

Figure 5.10 shows the same backward graph example, but with window size=3.

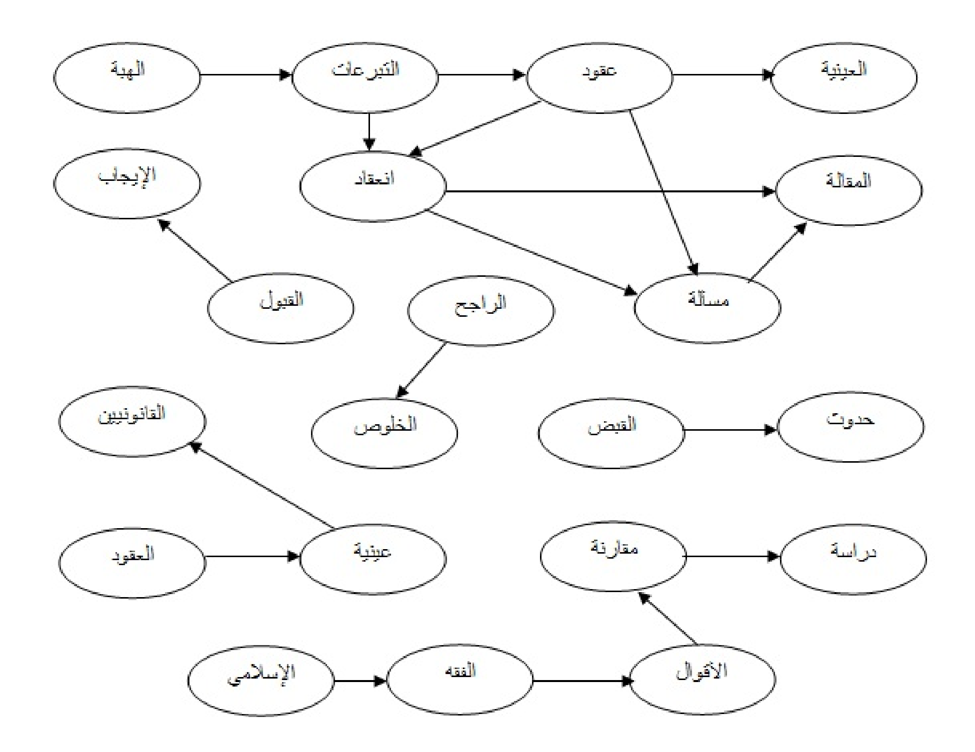
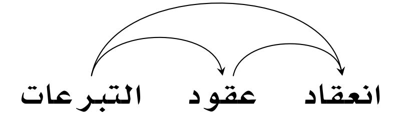


Figure 5.10: Backward Graph for the Document (42.ABSTR) with Window Size=3

If the window size is set to 3, the shaded tokens (vertices of the constructed graph) by yellow in Figure 5.4 are tested again to find if there existed vertices within a window size=3 in the original document. Then there will be a backward relation between these vertices according to their locations in the original document. For example the words (انعقاد), (عقود) and (التبرعات) are adjacents and selected as vertices in the constructed graph, so there will be a backward relation between them.

- (التبرعات) will point to (عقود)
- (التبرعات) will point to (انعقاد)
- (عقود) will point to (انعقاد)

As shown below:



5.3.3 Applying Page Rank Algorithm

After vertices that added to the constructed graph are restricted with syntactic filters. Which select only lexical units of a certain part of speech, only nouns of type (N or NT)

for each Arabic document (abstract and its title). All lexical units that pass the syntactic filter are added to the graph, and edges are drawn between those lexical units that co-occur within a window of words. After the graph is constructed (Bidirectional, Forward or Backward). The score associated with each vertex is set to an initial value of 1. And the page rank Equation 3.1 described in Section 3 is run on the graph vertices for several iterations until it converges – usually for 20-30 iterations, at a threshold of 0.0001. As mentioned in Section 3.3 in Equation 3.2 convergence is achieved when the error rate for any vertex in the graph falls below that threshold in this research. Where this error rate is approximated with the difference between two successive scores for graph vertices.

Once a final score is obtained for each vertex in the graph in a document, vertices are sorted in reversed order of their page rank scores. And the top 30% of the total tokens existed in each document are retrieved for post-processing steps for keywords and keyphrases extraction in a page rank list.

In this research, this percentage was used after applying several experiments on the training test and comparing them to reach the best possible rate that will lead to the best results. Each keyword is expanded into a set of keyphrases by searching for each occurrence of the token in the original text, and for each occurrence collecting all adjacent keywords and concatenating them into a keyphrase as will explained in the next section.

The top 30% words (17 words) from the document (42.ABSTR) that is shown in Figure 3.2 are retrieved for the post-processing stage in this system as shown in Equation 3.3.

$$Keywords = \lceil 30\% * TotalTokens \rceil \tag{5.1}$$

Where TotalTokens refers to the total number of tokens in the document. This number is equal to 55 in the document (42.ABSTR). Accordiing to Equation 5.1 number of selected vertices =17.

$$Keywords = \lceil 30\% * 55 \rceil$$

$$Keywords = \lceil 16.5 \rceil$$

$$Keywords = 17$$

5.3.4 Post-Processing Steps to Extract Keywords and Keyphrases

After removing stopwords, applying the above syntactic filters that select only lexical units of a certain part of speech, assigning a page rank score for each vertex in the constructed graph, arranging vertices in descending order according to their page rank scores and choosing the top 30% of the total number of words in the document as keywords. The post-processing stage to extract keywords and keyphrases can be detailed as follows:

1. Scan the original document token by token.
2. Add all single tokens found in the text and returned by the page rank algorithm to a candidate list of keyphrases with the corresponding page rank score.
3. Add all successive tokens found in the original text as a keyphrase to the keyphrases list with the corresponding page rank score which is equal to the summation of tokens page rank scores constituting that keyphrase.
4. Ignore all keyphrases of size greater or equal to 4 tokens.
5. Remove all keyphrases of size 3 (that consists of 3 tokens) if it has a candidate sub-keyphrase of size 2 token.
6. Sort all the returned keywords and keyphrases based on their page rank scores, then:
   (a) Add the top 3 keyphrases of size 1 (one word) to the final list of keyphrases.
   (b) Add the top 7 keyphrases of size 2 (two words) to the final list of keyphrases.
   (c) Add the top 2 keyphrases of size 3 (three words) to the final list of keyphrases.

To summarize the above steps, all tokens in the original document are scanned. If there existed two or more successive keywords then a keyphrase will be compoosed of these successive keywords. Else a keyword of a single word is selected.

In the proposed system keywords and keyphrases with several lengths are composed. After finding many statisticals for the used dataset, it has been found that the percentage of actual keyphrases with lengths 1, 2 and 3 words constitutes the greatest proportion of the total. It constitutes about 96.54% as will explained later in details in Chapter 6. So in the proposed system automatic keyphrases of sizes 1, 2 and 3 words are selected and automatic keyphrases with lengths greater than 3 words are ignored.

Finally specific number for the retrieved automatic keywords and keyphrases of lengths 2 or 3 words are selected according to some statisticals on the used data that will be also detailed later in Chapter 6. The top 3 keywords, the top 7 keyphrases of length=2 words and the top 2 keyphrases of length=3 words are selected as the extracted keywords and

keyphrases.

If the previous document (42.ABSTR) is taken using forward graph and window size 2 for an example. Tokens in the documnet are checked if they are existed in the generated keywords that are shown in Table 3.3. Using the previous steps as a post-processing:

1. Scan the document (42.ABSTR) token by token.

2. Add all the tokens in Table 3.3 that are generated by the page rank algorithm with their page rank scores to the keyphrases list.

3. Add the following successive tokens in the original text as keyphrases (عقود التبرعات), (مسألة انعقاد عقود التبرعات), (حدوث القبض), (عينية العقود) and (الفقه الإسلامي). With their page rank scores that are equal to the summation of the constituting tokens page rank scores.

4. Ignore the keyphrase (مسألة انعقاد عقود التبرعات) of size 4 tokens. No sub-keyphrases of size 2 are existed in keyphrases of size 3.

5. Sort all the returned keywords and keyphrases based on their page rank scores, then:

   (a) Add the top 3 keyphrases of size 1 (one word) to the final list of keyphrases: (الهبة), (الراجح) and (الإيجاب).

   (b) Add the top 7 keyphrases of size 2 (two words) to the final list of keyphrases: (عقود التبرعات), (عينية العقود) and (حدوث القبض). Since there existed just 3 keyphrases of size 2.

   (c) Add the top 2 keyphrases of size 3 (three words) to the final list of keyphrases. No keyphrases of size 3 will be added since there are not such generated keyphrases of size 3.

The expected results for the proposed system are a set of words and phrases that are representative for a given document. Based on the used graph type (Forward, Backward or Bidirectional) and the used window size.

5.3.5 Weighted Vertices in Graphs

In this research, two methods were used as weights for the graph vertices:

- The final page rank score for each token is multiplied by its frequency in the documnet.

- Each page rank score for the selected keyword that is existed in the document title is multiplied by a weight number that is determined by (3). After applying several experiments as will shown later in chapter 6.

Experiment results showed that using these weights for graph vertices has improved the results significantly. As illustrated in section 3.4.4, vertices are connected through links (edges). These links may be strong or poor depending on several factors. For example, words may occur more than once in the document. Or there exist some words that having greater pagerank scores pointing to other words. These links between such words are more strong, which means a higher pagerank score may be obtained. This edges weight that is used by Mihalcea and Tarau in their paper. In this proposed system other types of weights are used as mentioned above. Weights for vertices in the constructed graph are used according to the term (word) frequency in the document and its position with in the documnet, either in the title or the abstract.

Chapter 6

Experimental Results

6.1 Introduction

This chapter details the experiments results of keywords and keyphrases extraction system using TextRank adapted from Mihalcea and Tarau on Arabic dataset. Many experiments with different sizes of windows were implemented on the training dataset to choose the best one according to the highest value of F-Measure. Then apply it on the testing dataset to be adopted as the final results.

6.2 Dataset

In this research, text rank algorithm was tested on a collection of Arabic published articles that were collected manually from the Internet from different disciplines . This dataset consists of (150) Arabic abstracts and their titles with their corresponding keywords that were built manually and included in the articles. It was divided into 2 sets: 100 for training, 50 for testing.

The total number of tokens in all these abstracts is 21704, with an average of 144.69. The smallest abstract has 36 tokens, while the longest one has 314 tokens.

Some statistics were calculated and concluded from this dataset which had a great benefit in the experiments implementation. As mentioned in Section 3.5 Arabic abstracts with their titles were reprinted in Notepad files, and their corresponding keywords were also reprinted in Notepad files with the same names of abstract files but with different extensions. Every file has its own number of keywords and keyphrases that were built manually.

In order to determine the number keyphrase in all documents, a full statistics are applied for these keyphrases. Table 6.1 shows these statistics. It shows how many documents

have the specific number of keyphrases.

Table 6.1: Frequencies of Keyphrases in the Collected Dataset

Number of Keyphrases	Total No. of Documents
2	8
3	63
4	41
5	23
6	7
8	6
10	1
12	1

It can be noticed from Table 6.1 that the minimun number of keyphrases that found in all documents is 2 keyphrases, while the maximum number is 12.

Keyphrases may consist of one word or more than one. They may consist of two, three, four or more words. In order to determine the length of each keyphrase, a full statistics are applied also for these keyphrases. Table 6.2 shows the number of existed and non existed keyphrases in all documents.

Table 6.2: Number of Existed and Non Existed Keyphrases in Documents According to their Numbers

No. of keyphrases	No. of Documents	Total No. of Keyphrases	No. of Existed Keyphrases	No. of Non-Existed Keyphrases
2	8	16	10	6
3	63	189	81	108
4	41	164	49	115
5	23	115	26	89
6	7	42	5	37
8	6	48	4	44
10	1	10	0	10
12	1	12	0	12

The third column in Table 6.2 is calculated by multiplying the first column by the second one.

Since the page rank algorithm idea in this research totally depends on keyphrases that are existed in the abstracts with their titles (documents). Another statistic was calculated from documents and their actual keyphrases that shows how many keyphrases with different lengths are existed in these documents. This is due that page rank algorithm depends on the co-occurrence relations between the existed words in the documents.

Keyphrases are checked for different situations (search places):

1. Document title
2. Document abstract
3. Document title or abstract
4. Document title and abstract

Table 6.3 shows the number of existed and non existed keyphrases from different search places as it has been explained above. Statistics in table 6.3 have a great benefit in the weighted pagerank equation. Results will be improved when using suitable weights in the pagerank equation. For example, in this research, automatic keywords and keyphrases that are found in titles were given more weights. Because it has been noticed that most keyphrases are existed in titles. Also it shows the maximum expected recall for each situation which is equal by dividing the total exiested keywords and keyphrases by the total ones (596) that are found manually from the used dataset. Table 6.3 also shows the maximun recall that can be reached when searching the documents (Abstracts and their Titles) which is 63%. It means that the ideal recall that can be got is 63% and not 100% since not all actual keywords and keyphrases are existed in the documents.

Table 6.3: Number of Existed Keyphrases with Different Lengths (in Words) from Different Search Places

Search Place	**1 Word**	**2 Words**	**3 Words**	**4 Words**	**5 Words**	**6 Words**	**Recall**
Title	61	167	33	7	3	0	45.46%
abstract	80	192	38	8	3	1	54.02%
Title or Abstract	**90**	**227**	**46**	**9**	**3**	**1**	**63.08%**
Title and Abstract	51	132	25	6	3	0	36.40%

It can be noticed from Table 6.3 that the minimum keyphrase length (in words) is 1, while the maximun keyphrase length (in words) is 6. Using title or abstarct (document)

search place results, the mechanism that has been followed to determine the number of retrieved keyphrases (postprocessing steps as shown in Chapter 3) based on their lengths is as follows:

1. The total existed keyphrases in abstracts and their titles is 376 from the total 596.
2. Determine the percentage of retrieved keywords to be 30% from the total words in a document.
3. Determine the number of retrieved keyphrases to be 12 since the maximun number of keyphrases in a documnet is 12 as shown in Table 6.1.
4. Focus on keyphrases that consist of one, two and three words because they constitute 96.54% of the total existed keyphrases.
5. Determine the number of keyphrases of one word (Single Word) according to the following equation:

$$\left\lceil 12 * \frac{90}{376} \right\rceil = 3$$

   where 12 is the total keyphrases to be retrieved, 90 is the number of existed keyphrases of one word in documents and 376 is the total number of existed keyphrases in documents.

6. Determine the number of keyphrases of two words according to the following equation:

$$\left\lceil 12 * \frac{227}{376} \right\rceil = 7$$

   where 227 is the number of existed keyphrases of two words in documents.

7. Determine the number of keyphrases of three words according to the following equation:

$$\left\lceil 12 * \frac{46}{376} \right\rceil = 2$$

   where 46 is the number of existed keyphrases of three words in documents.

So the number automatic kephphrases to be retrieved is 12 that will be divided as follows: 3 keyphrases that consist of one word, 7 keyphrases consist of two words and 2 keyphrases consist of three words.

6.2.1 Training and Testing Datasets

Separating data into training and testing sets is an important part of evaluating Keywords Extraction Systems . Typically, when a dataset is separated into a training set and testing set, most of the data is used for training, and a smaller part of the data is used for testing. Testing and training sets are randomly selected to ensure that they are similar. By using similar data for training and testing, the effects of data discrepancies can be minimized and the characteristics of the system are better understood (Microsoft, 2013).

After a system has been trained by using the training set, the system is tested by making predictions against the testing set. Because the data in the testing set already contains known values for the attribute that you want to predict, it is easy to determine whether the system's guesses are correct.

In this research, the Arabic dataset is randomely sampled into (100) documents for training and (50) documents for testing. The random division process is implemented by a Perl programm into a set of training and testing documents without human intervention.

6.3 Experiments Setup

In this proposed system, many features are added to the original page rank algorithm, because of the theory that they would improve the results.

The following is a list of the features that are tested to check if they improved the results:

1. Graph Type:
   Which graph type is suitable to represent the document words (Bidirectional, Forward or Backward).

2. Stopwords removal.

3. Part-of-speech tagging tagging.

4. Window size (2, 3, 5 or 10)

5. Using weights for vertices.

All different combinations of the above features are tested to find which feature that may improve the results.

6.3.1 Graph Type

The first experiments were performed based on the graph edge direction including stop-words and without part of speech tagging process for the tokens in the documents. Table 6.4 shows the results when applying keywords and keyphrases extraction system using page rank algorithm using bidirectional graph with window sizes 2, 3, 5 and 10 on training set.

Table 6.4: TextRank using Bidirectional Graph (BI) varying Window Size (w) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
BI, w=2	401	1142	12	0.0352	0.0110	0.0162
BI, w=3	401	1142	12	0.0352	0.0110	0.0162
BI, w=5	401	1142	12	0.0352	0.0110	0.0162
BI, w=10	401	1142	12	0.0352	0.0110	0.0162

Table 6.5 shows the results when applying the system using forward graph also with the above window sizes 2, 3, 5 and 10.

Table 6.5: TextRank using Forward Graph (FD) varying Window Size (w) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, w=2	401	1074	8	0.0196	0.0071	0.0101
FD, w=3	401	1079	9	0.0236	0.0089	0.0126
FD, w=5	401	1073	10	0.0240	0.0114	0.0145
FD, w=10	401	1043	15	0.0363	0.0180	0.0209

Table 6.6 shows the results when applying the system using backward graph with window sizes 2, 3, 5 and 10.

Table 6.6: TextRank using Backward Graph (BD) varying Window Size (w) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
BD, w=2	401	1095	2	0.0058	0.0017	0.0027
BD, w=3	401	1088	2	0.0050	0.0019	0.0028
BD, w=5	401	1078	7	0.0203	0.0067	0.0100
BD, w=10	401	1051	10	0.0248	0.0119	0.0146

Based on the above results and according to the F-measure, forward graph can be adopted instead of backward and bidirectional graphs. Since it has the highest F-measure values in all window sizes. So the rest of experiments will be performed on forward graph. Figure 6.1 shows the above experiment with different window sizes 2, 3, 5 and 10 using bidirectional (BI), forward (FD) and backward (BD) graphs on training set.

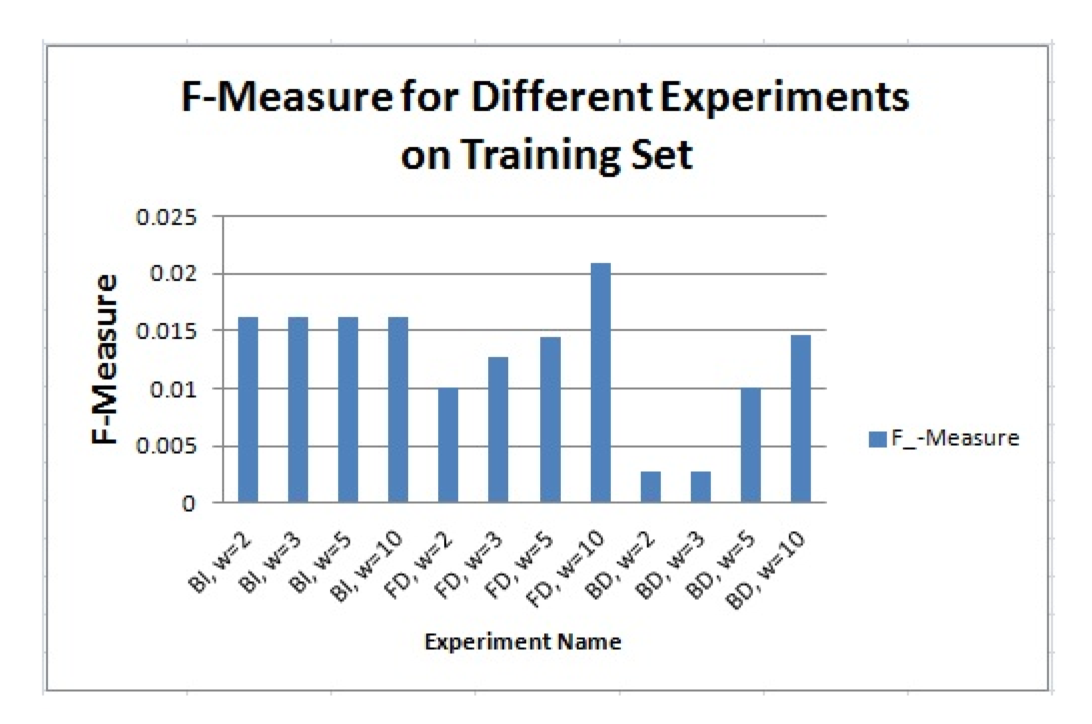


Figure 6.1: F-measure for Different Experiments using Bidirectional (BI), Forward (FD) and Backward (BD) graphs with Different Window Sizes on Training Set

All experiments results on training set using forward, backward or bidirectional graphs are existed in Appendix B.

6.3.2 Stopwords Removal Experiments

The second part of experiments is to remove stopwords from the training set then apply the proposed system using forward graph, but without tagging process with window sizes 2, 3, 5 and 10. Table 6.7 shows the results after removing stopwords then applying the system using forward graph with the above window sizes.

Table 6.7: TextRank using Forward Graph (FD) after removing Stopwords and varying Window Size (w) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, w=2	401	1141	47	0.1259	0.0396	0.0589
FD, w=3	401	1131	53	0.1464	0.0456	0.0682
FD, w=5	401	1139	58	0.1605	0.0519	0.0754
FD, w=10	401	1135	63	0.1750	0.0566	0.0827

As noted from the Results, stopwords removal from the documents as a preprocessing before applying the proposed system will lead to better results. Since stopwords are very common words that appear in the text that carry little meaning; they serve only a syntactic function but do not indicate subject matter. They can cause problems when searching for phrases that can best describe the document, so removing stopwords will yield to better results. Because the competition will be just between the needed tokens. Figure 6.2 shows the above experiment with different window sizes 2, 3, 5 and 10 using forward graph with stopwords and after stopwords removal on training set.

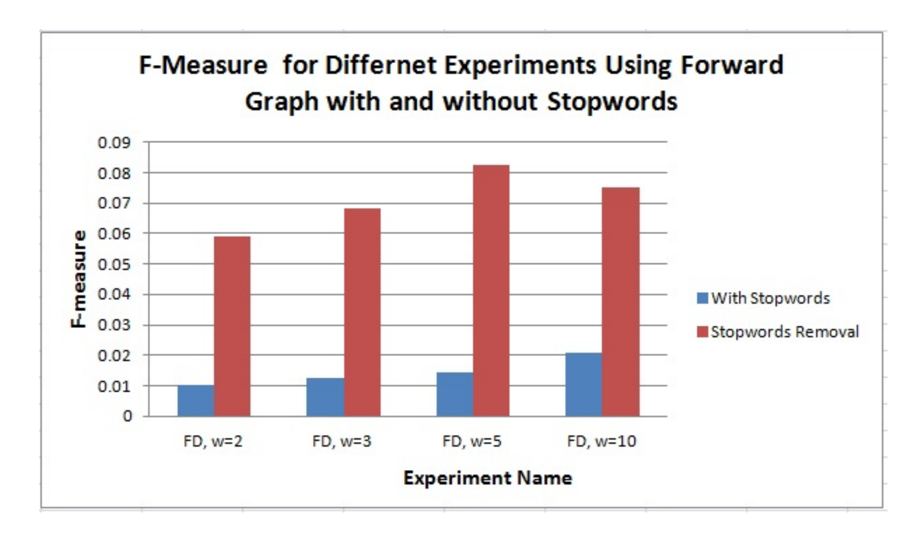


Figure 6.2: F-measure for Different Experiments using Forward graph with Different Window Sizes with Stopwords and after Stopwords Removal on Training Set

All experiments results on training set using forward, backward or bidirectional graphs either before or after removing stopwords are existed in Appendix B.

6.3.3 Linguistic Features: Part of Speech Experiments

The third part of experiments after removing stopwords from the training set is applying the linguistic feature words part of speech tagging on the documents. Then apply the proposed system using forward graph with window sizes 2, 3, 5 and 10. Table 6.8 shows the results after removing stopwords, tokenizing words in documents by tagging process then applying the proposed system using forward graph with window sizes 2, 3, 5 and 10.

Table 6.8: TextRank using Forward Graph (FD) after removing Stopwords, with POS tagging and varying Window Size (w) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, w=2	401	982	107	0.2785	0.1093	0.1535
FD, w=3	401	987	104	0.2710	0.1055	0.1478
FD, w=5	401	986	101	0.2662	0.1028	0.1445
FD, w=10	401	989	103	0.2687	0.1048	0.1471

Figure 6.3 shows the above experiment with different window sizes 2, 3, 5 and 10 using forward graph after removing stopwords with POS tagging (With) and without POS tagging (Without) on training set.

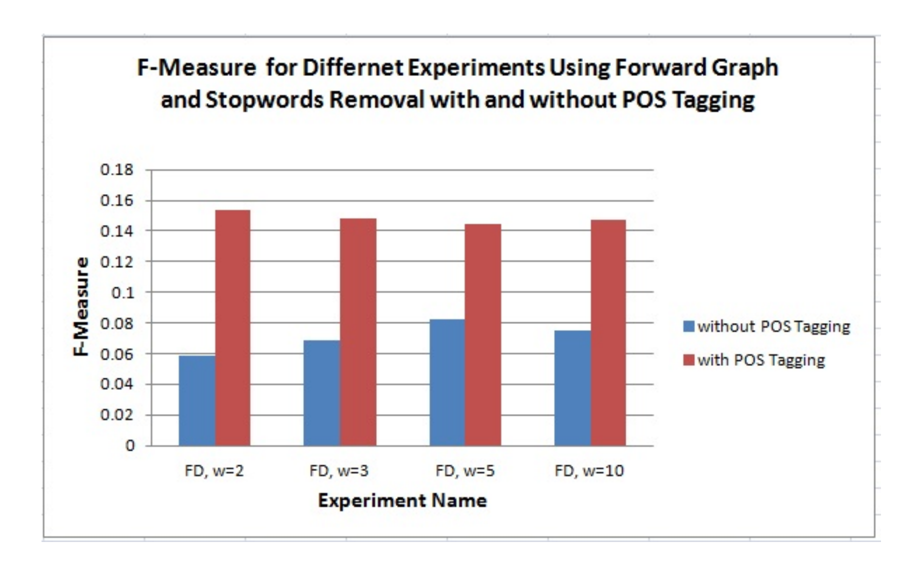


Figure 6.3: F-measure for Different Experiments using Forward graph with Different Window Sizes after removing Stopwords with POS tagging and without POS tagging on Training Set

As summary from the previous table, the following is a list of features that significantly improve the accuracy of the proposed system using the training dataset:

1. Using forward graph;
2. Applying stopwords removal;
3. Applying POS tagging;
4. Using the window size 2.

In order to check the performance of the proposed system using the testing dataset, the above features are applied. Table 6.9 shows the different measures for the testing dataset.

Table 6.9: TextRank using Forward Graph (FD) after removing Stopwords, with Tagging and Window Size=2 on Testing Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, w=2	195	493	49	0.2855	0.1014	0.1473

All experiments results on training set using forward, backward or bidirectional graphs either before or after removing stopwords, and either with or without tagging are existed in Appendix B.

6.3.4 Using Weights for Vertices in the Graph

The final part of experiments is to add certain weights to the page rank scores for each word based on several factors. Numerous experiments were applied to get the suitable weights that will improve the results. Three types of weights are used: Token frequency, token position (abstract or title) and the combination between them.

6.3.4.1 Token Frequency as a Weight

In order to check the effect of a token frequency for the training and testing datasets, each page rank score for each word is multiplied by its frequency in a document. In Arabic texts words that have more frequencies must be given more weights to be nominated as keywords. Table 6.10 shows the final results after this process is done for the training dataset. The F-measure score has significantly increased from 0.1535 to 0.1683 on training set.

Table 6.10: TextRank using Term Frequency in Forward Graph (FD) after removing Stopwords and with tagging varying Window Size (2) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, w=2	401	982	117	0.3060	0.1203	0.1683

Table 6.11 shows how using term frequency as a weight in the proposed system has also improved the results on testing set. The F-measure score has significantly increased from 0.1473 to 0.1560 on testing set.

Table 6.11: TextRank using Term Frequency in Forward Graph (FD) after removing Stopwords and with tagging varying Window Size (2) on Testing Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, w=2	195	493	52	0.3032	0.1072	0.1560

6.3.4.2 Term Position as a Weight

It has noted that automatic keywords and keyphrases generated by the proposed system and existed in Titles are more nominated to be adopted and given more weights. Every generated keyword or keyphrase existed in the title is mulitiplied by a number to give it more weight. In this research, keywords and keyphrases page rank scores existed in titles are multiplied by 1, 2, 3, 5, 10, 15, 20 and 30 to test which weight number can lead to the better results. Table 6.12 shows how using term position as a weight in the proposed system with the above weights using forward graph and window size=2 on the training set.

Table 6.12: TextRank using Several Term Position Weights (WT) in Forward Graph (FD) after Removing Stopwords, with POS Tagging, and varying Window Size (2) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, WT=1	401	982	107	0.2785	0.1093	0.1535
FD, WT=2	401	983	112	0.2912	0.1139	0.1601
FD, WT=3	401	983	113	0.2945	0.1148	0.1615
FD, WT=4	401	984	112	0.2912	0.1137	0.1598
FD, WT=5	401	984	112	0.2912	0.1137	0.1598
Fd, WT=10	401	984	112	0.2912	0.1137	0.1598
FD, WT=15	401	984	112	0.2912	0.1137	0.1598
FD, WT=20	401	984	112	0.2912	0.1137	0.1598
FD, WT=30	401	984	112	0.2912	0.1137	0.1598

As noted from Table 6.14, that when increasing the weight number the F-measure is increased. Until values will be fixed and unchangable when increasing weight number. The highest value of f-measure will be at weight number=3. So this weight will be always used and multiplied by tokens that existed in titles. Table 6.13 shows how using term position as a weight in the proposed system with the above weights on testing set.

Table 6.13: TextRank using Several Term Position Weights (WT) in Forward Graph (FD) after Removing Stopwords, with POS Tagging, and varying Window Size (2) on Testing Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, WT=1	195	493	49	0.2855	0.1014	0.1473
FD, WT=2	195	493	53	0.3078	0.1108	0.1604
FD, WT=3	195	493	53	0.3088	0.1112	0.1610
FD, WT=4	195	494	53	0.3088	0.1107	0.1605
FD, WT=5	195	494	53	0.3088	0.1107	0.1605
FD, WT=10	195	494	53	0.3088	0.1107	0.1605
FD, WT=15	195	494	53	0.3088	0.1107	0.1605
FD, WT=20	195	494	53	0.3088	0.1107	0.1605
FD, WT=30	195	494	53	0.3088	0.1107	0.1605

The same results have appeared when using several weights for tokens in titles. The highest value of f-measure will be at weight number=3.

6.3.4.3 Term Frequency and Term Position as a Combined Weight

When using term frequency weight combined with term position weight, results will be improved based on the F-Measure value. Table 6.14 shows how using the term frequency weight and term position weights together using forward graph, after stopwords romoval, with POS tagging and window size=2 on training set. Using the weight number=3 to multiply it with tokens that are existed in titles.

Table 6.14: TextRank using Term Frequency and Position Weight (WT), Forward Graph (FD), after Stopwords Removal, with POS Tagging and Window Size (2) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, WT=3	401	684	111	0.2948	0.1936	0.2102

Table 6.15 shows how also using the above weights in the proposed system has improved the results on testing set using forward graph, stopwords removal, words tagging with window size=2 and weight number=5 on testing set.

Table 6.15: TextRank using Term Frequency and Position Weight (WT), Forward Graph (FD), after Stopwords Removal, with POS Tagging and Window Size (2) on Testing Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, WT=3	195	283	43	0.2465	0.1828	0.1982

Figure 6.4 shows the experiments results using forward graph, stopwords removal, POS tagging and window size=2 with the above weights on training set.

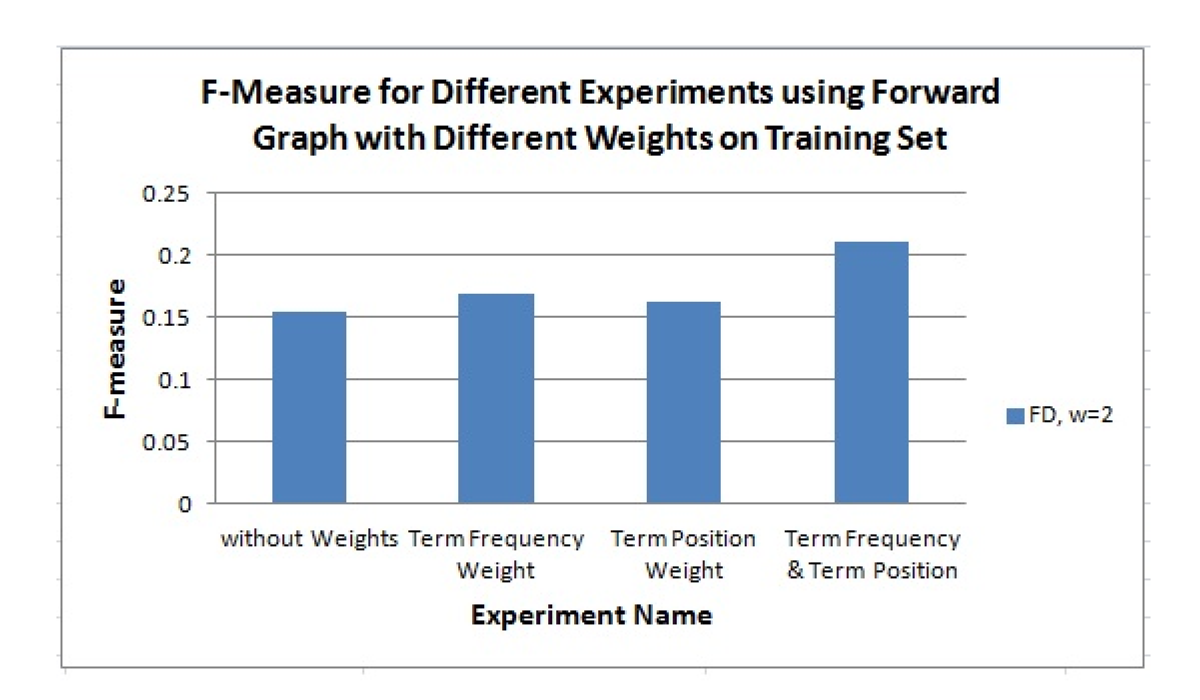


Figure 6.4: F-measure for Different Experiments using Forward graph, after removing Stopwords, with POS Tagging and Window Size=2 with Different Weights on Training Set

Figure 6.5 shows the experiments results using forward graph, stopwords removal, POS tagging and window size=2 with the above weights on testing.

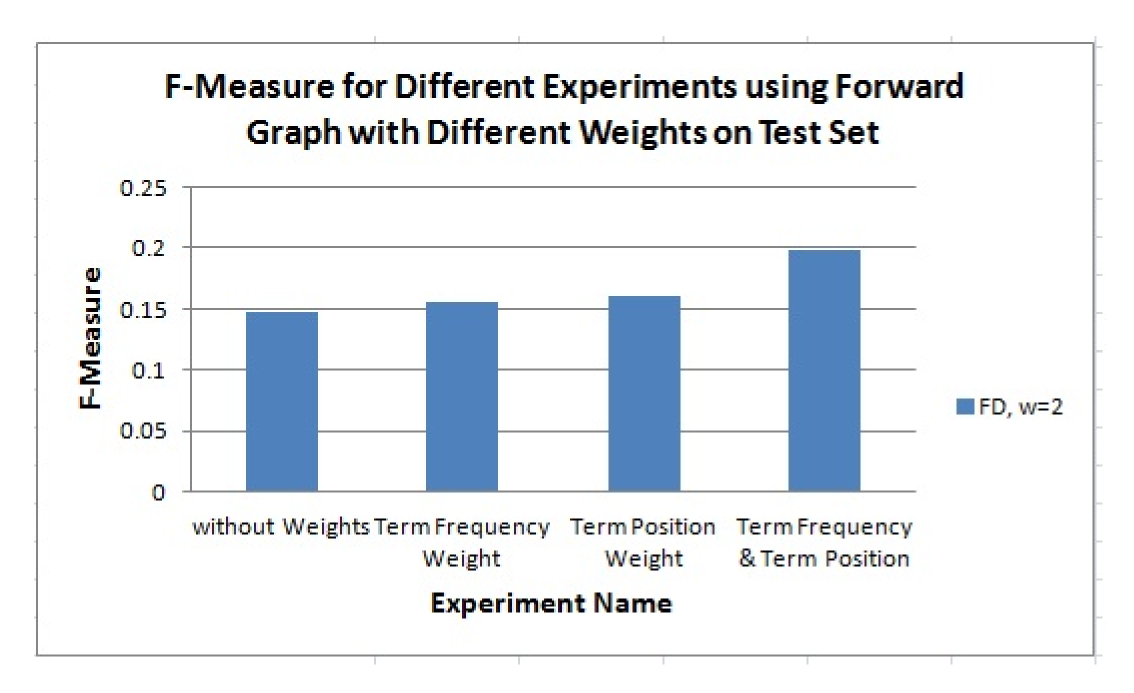


Figure 6.5: F-measure for Different Experiments using Forward graph, after removing Stopwords, with POS Tagging and Window Size=2 with Different Weights on Testing Set

As a result of applying several experiments on the training set the suitable situations and tools that lead to the best results are concluded as follows:

- Using forward graph
- Removing stopwords from documents as a preprocessing step.
- Applying the linguistic feature: part of speech tags on documents words as a preprocessing step.
- Using the window size = 2
- Multiply the graph vertices page rank scores by their frequencies and multiply the terms page rank scores existed in titles by 3.

Mihalcea and Tarau did not use such weights for vertices in graphs in their research. They did not benefit from words frequencies in documents. And they did not assign words that are existed in titles any weight or importance. They concentrated on in their model on the graphs that may include multiple or partial links between the units (vertices) that are extracted from text. It depends on edges weights that will indicate the strength of the connection between vertices (Mihalcea & Tarau, 2004). Figure 6.6 illustrates all the previous experiments on training set. That's where the blue columns represent applying the proposed system with stopwords, without tagging and without weights. The red columns represent applying the system without stopwors, without tagging and without weights. The green columns represent applying the system after removing stopwords, with tagging and without weights. And finally the purple columns represents applying the system after removing stopwords, with tagging and with weights. All these experiments were tested with several window sizes 2, 3, 5 and 10 on the training set.

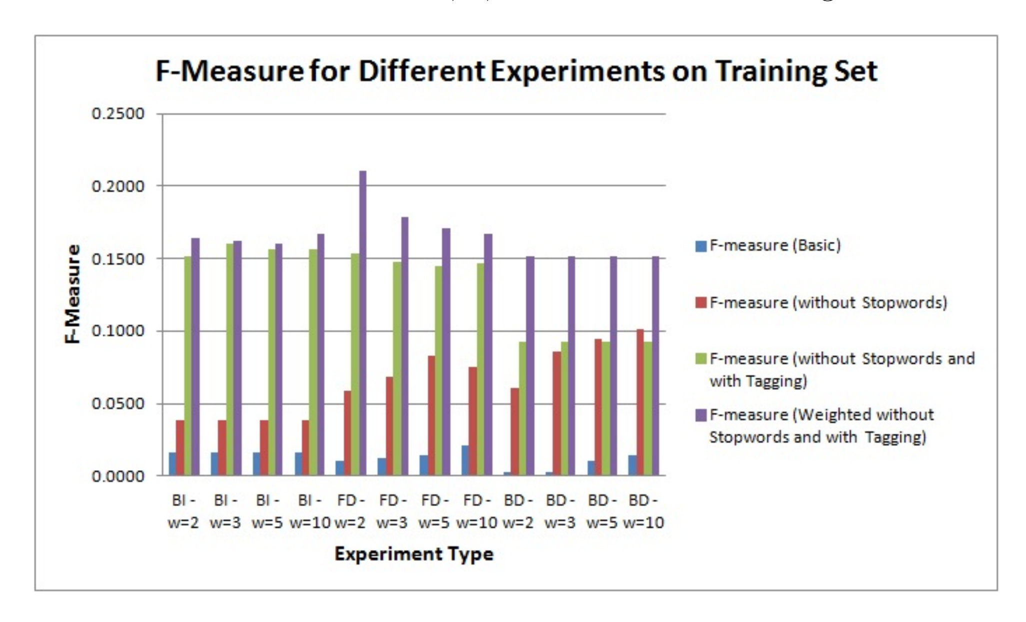


Figure 6.6: F-measure for Different Experiments on Training Set

Figure 6.7 shows the same experiments and results of Figure 6.6 but on the testing set.

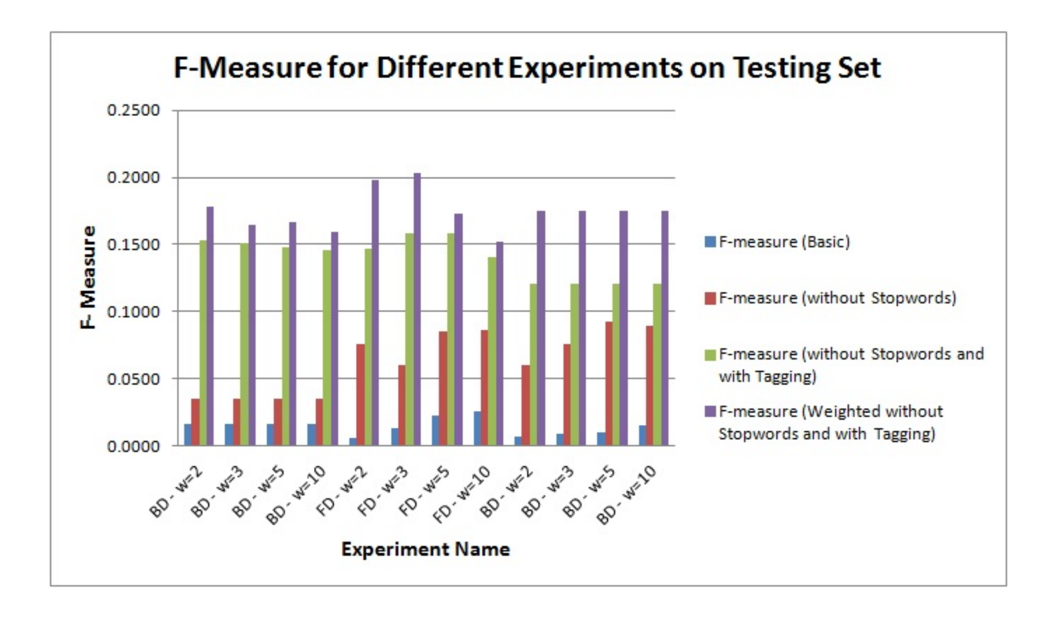


Figure 6.7: F-measure for Different Experiments on Testing Set

6.4 Comparison between the Original and the Modified Algorithm

The original algorithm for keywords and keyphrases extraction using page rank as proposed by (Mihalcea & Tarau, 2004) and explained in Section 3.4 on English text is applied on the Arabic training set. The results shown in Table 6.16 are obtained:

Table 6.16: Original TextRank using Forward Graph (FD) after removing Stopwords, with POS tagging and Window Size (2) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, w=2	401	982	107	0.2785	0.1093	0.1535

Table 6.17 shows the results after applying the original algorithm on the Arabic testing set.

Table 6.17: Original TextRank using Forward Graph (FD) after removing Stopwords, with Tagging and Window Size=2 on Testing Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, w=2	195	493	49	0.2855	0.1014	0.1473

After modifying the algorithm and adding the weights terms as explained in Section 6.3.4, the shown results in Table 6.18 are obtaind on the Arabic training set.

Table 6.18: Modified TextRank using Term Frequency and Position Weight (WT), Forward Graph (FD), after Stopwords Removal, with POS Tagging and Window Size (2) on Training Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, WT=3	401	684	111	0.2948	0.1936	0.2102

Table 6.19 shows the results after applying the modified algorithm on the Arabic testing set.

Table 6.19: Modified TextRank using Term Frequency and Position Weight (WT), Forward Graph (FD), after Stopwords Removal, with POS Tagging and Window Size (2) on Testing Set

Experiment Name	Relevants	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
FD, WT=3	195	283	43	0.2465	0.1828	0.1982

It can be noted from the above tables how using weights terms in the modified algorithm has improved the system obviously. It has increased the F-Measure from 0.1535 to 0.2102 on training set, and from 0.1473 to 0.1982 on testing set.

Chapter 7

Conclusion

7.1 Introduction

This chapter presents the conclusions of this research. A summary of the thesis is introduced. The study questions and the contributions of this study are discussed. And finally future works of the current system are presented.

7.2 Conclusion

The main goal of this research is to build a keywords and keyphrases extraction system using page rank algorithm for Arabic language. This system is based on Text Rank – a graph based ranking model for text processing. This research shows how it can be successfully to use the proposed system on Arabic text for natural language applications.

The evaluation methods have shown that the accuracy achieved by the proposed system on Arabic language is competitive. Keywords and keyphrases extraction system using page rank algorithm works well because it does not only rely on the local context of a word in a text (vertex), but rather it takes into account information recursively drawn from the entire text (graph).

The basic idea of the proposed system is summarized by constructing a network graph using candidate keywords as vertices, and co-occurrence to draw edges between them with in a certain window size. Edges can be forward, backward or bidirectional. Then runs the page rank algorithm upon the graph to rank each keyword's importance. Each keyword is expanded into a set of keyphrases by searching for each occurrence of the keyword in the original text, and for each occurrence collecting all adjacent words that are eligible keywords and concatenating them into a phrase. The output keyphrase page rank score is the sum of the composed keywords page rank scores.

Several experiments are performed on the training set to decide which situations and tools can lead to better results. The results are evaluated using precision, recall, and F-measure. The maximum recall that can be achieved on the testing set of this collection which consists of 50 Arabic documents is 63%. Since not all the manual keywords and keyphrases are existed in the articles' abstracts and their titles. This proposed system achieved 25% of recall which is acceptable and competitive compared with the recall value that has been achieved on English testing collection which consists of 500 English documents by Mihalcea and Tarau, They have achieved 42% of recall where the ideal recall of their testing set was 78%. Despite of the difficulties and challenges of the Arabic language and using less number of documents in the Arabic testing collection than English.

The first experiments are tested by applying the basic steps of the proposed system. That consist of constructing a graph for each document by adding all the words in this document as vertices to that graph, and then build edges according to the co-occurrence relation between these vertices (words). Finally apply the page rank algorithm on the grapgh vertices acoording to the edges types, numbers and window size until convergence is occurred. This basic idea is performed using forward, backward and bidirectional graphs.

The results of the proposed system on the chosen training set have shown that using forward graph led to the best results. With the stability of the system to use forward graph other experiments. Stopwords are removed from all documents as a preprocessing step before constructing the graph. This step has significantly improved the results. After that, words part of speech (POS) tagging is tested by other experiments and also results have been clearly improved, linguistic feature such as part of speech tags shows that it is very helpful in keywords and keyphrases extraction task. Then results with window size 2 were noted as the best one. Finally certain weights based on word position (Abstract or Title) and word frequency were added to the page rank scores of keywords and keyphrases in every document which has had a significant role in improving results significantly.

7.3 Answering the Research Questions

This research has proved that it is possible to design a keywords and keyphrases extraction system using page rank algorithm on Arabic text as from English text.

Arabic language has been given little interest compared to other language. Researches on Arabic language is still in its infancy. In the last few years, researches of keywords extraction have focused mostly on English language, but from Arabic text is still rarely

applied due to the difficulty of the Arabic language. There are many challenges of the Arabic language, It is both morphologically rich and highly ambiguous. It has complex morpho-syntactic agreement rules and a lot of irregular forms.

The reseach questions were:

- Is it possible to extract keywords from Arabic texts using page rank algorithm as from English texts despite of Arabic Language difficulties?.

  The experiments results have shown that it is possible to build a keywords extraction system using page rank algorithm from Arabic texts. The main bjective of this thesis is as follows:

  - To design a Keywords and keyphrases extraction system using page rank algorithm from Arabic texts that can be competitive.

- Are the linguistic features such as part of speech help to improve the performance of keywords and keyphrases extraction system?.

  Also the experiments results have shown that removing all meaningless stopwords, and nominating general nouns that are existed either in the abstracts or titles will improve the results significantly.

- Is the word position feature (in title or abstract) helps to improve the performance of keywords and keyphrases extraction?.

  The experiment results have proved that word position is an important factor. When assigning words that are existed in titles more weights, this will improve the results also significantly.

- Does the word frequency feature in each document help to improve the performance of keywords and keyphrases extraction?.

  The experiment results have proved that word frequency in each document is also an important factor. When multiplying each word page rank score with its frequency, this will improve the results also significantly.

7.4 Future Work

This research can be further developed and improved in a number of directions as follows:

- Improving the system to be applied on more documents by increasing dataset size.

- Applying differnt weighting schemes on the page rank algorithm that could be useful for improving results.
- Applying the keywords and keyphrases extraction using page rank algorithm on documents that have larger lengths than just abstracts.
- Applying the keywords and keyphrases extraction using page rank algorithm on multi documents of the same topic.

References

Al-Hamad, A., & Al-Zoubi, Y. (1993). Al-moujam al-wafi fi adawat al-nahw al-arabi. Dar Al-Amal.

Al-Hashemi, R. (2010). Text summarization extraction system (tses) using extracted keywords. *International Arab Journal of e-Technology*, *1*(4), 164-168.

Al-Muhtaseb, H., & Mellish, C. (1998). Some differences between arabic and english: A step towards an arabic upper model. In *Proceedings of the 6th international conference on multilingual computing* (p. 111-121). Cambridge, UK.

AL-Shalabi, R., Kanaan, G., & Gharaibeh, M. (2006). Arabic text categorization using knn algorithm. In *The 4th international multiconference on computer science and information technology* (Vol. 4).

Belkredim, F., & El-Sebai, A. (2009). An ontology based formalism for the arabic language using verbs and their derivatives. *Communications of the IBIMA*, *11*, 1943-7765.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hyper-textual web search engine. *Computer Networks and ISDN Systems*, *30*, 107-117.

Cohen, J. (1995). Language and domain-independent automatic indexing terms for abstracting. *Journal of the American Society for Information Science*, *46*(3), 162-174.

Creighton. (2013). *http://www.creighton.edu/fileadmin/user/hsl/docs/ref/searching_-_recall_precision.pdf.*

El-Beltagy, S., & Rafea, A. (2009). Kp-miner: A keyphrase extraction system for english and arabic documents. *Journal of Information Systems*, *34*(1), 132-144.

El-Hadj, Y., Al-Sughayeir, I., & Al-Ansari, A. (2009). Arabic part-of-speech tagging using the sentence structure. In *Proceedings of the second international conference on arabic language resources and tools.*

El-shishtawy, T., & Al-sammak, A. (2009). Arabic keyphrase extraction using linguistic knowledge and machine learning techniques. In *Proceedings of the second international conference on arabic language resources and tools* (Vol. 1).

El-Shishtawy, T., & El-Ghannam, F. (2012). Keyphrase based arabic summarizer (kpas). In *The 8th infos2012 international conference on informatics and systems* (Vol. 4).

Ercan, G., & Cicekli, I. (2007). Using lexical chains for keyword extraction. *Information Processing Management*, *43*(6), 1705-1714.

Frank, E., Paynter, G., Witten, I., Gutwin, C., & et al. (1999). Domain-specific keyphrase extraction. In *Proc. sixteenth international joint conference on artificial intelligence* (Vol. 31, p. 668-673).

Gupta, V., & Lehal, G. (2011). Automatic keywords extraction for punjabi language. *IJCSI International Journal of Computer Science*, *8*(3).

Habash, N., Soudi, A., & Buckwalter, T. (2007). Arabic computational morphology, knowledge-based and empirical methods. In (chap. 2). Springer Netherlands.

Hammadi, O., & Aziz, M. (2012). Grammatical relation extraction in arabic language. *Journal of Computer Science*, *8*(6), 891-898.

Hu, X., & Wu, B. (2006). Automatic keyword extraction using linguistic features. In *Data mining workshops* (p. 19-23).

Hulth, A. (2003a). Improved automatic keyword extraction given more linguistic knowledge. In *Conference on emprical methods in natural language processing* (p. 216-223).

Hulth, A. (2003b). Textrank: Bringing order into texts. In *Proceedings of the 2003 conference on empirical methods in natural language processing* (p. 216-223). Japan.

Insight, M. (2013). *http://www.mathinsight.org/undirected_graph_definition.*

Jiao, H., Liu, Q., & bo Jia, H. (2007). Chinese keyword extraction based on n-gram and word co-occurrence. In *Computational intelligence and security workshops* (p. 152-155).

Kouninef, B., & AL-Johar, B. (2011). Extracting entities and relationships from arabic text for information system. *Journal of Emerging Trends in Computing and Information Sciences*, *2*(11), 641-645.

Krishnan, M., Banerjee, S., Chakraborty, C., & Chakraborty, C. (2010). Statistical analysis of mammographic features and its classification using support vector machine. *Expert Systems with Applications*, *37*(6), 470-478.

Lawrence, P., Sergey, B., Motwani, R., & Winograd, T. (1998). *The pagerank citation ranking: Bringing order to the web* (Tech. Rep.). Stanford University.

Liu, F., Pennell, D., Liu, F., & Liu, Y. (2009). Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Human language technologies: The 2009 annual conference of the north american chapter of the acl* (p. 620–628). Boulder, Colorado.

Luhn, H. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, *1*(4), 309-317.

Mahgoub, H., Rösner, D., Ismail, N., & Torkey, F. (2008). A text mining technique using association rules extraction. *International Journal of Computational Intelligence*, *4*(1), 21-28.

Manning, C., & Schutze, H. (2000). Foundations of statistical natural language processing. In (chap. 8). The MIT Press Cambridge, Massachusetts London, England: Foundations of Statistical Natural Language Processing.

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, *19*(2), 313-330.

Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, *13*(4).

Microsoft. (2013). *http://msdn.microsoft.com/en-us/library/bb895173.aspx.*

Mihalcea, R., Liu, H., & Lieberman, H. (2006). Nlp (natural language processing) for nlp (natural language programming). In *7th international conference, cicling* (Vol. 3878, p. 319-330).

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into texts. In *Proceedings of empirical methods on natural language processing conference.*

Muaidi, H. (2008). *Extraction of arabic word roots: An approach based on computational model and multi-backpropagation neural networks.* Unpublished doctoral dissertation, university of De Montfort.

Oelze, I. (2009). *Automatic keyword extraction for database search.*

Plas, L., Pallotta, V., Rajman, M., & Ghorbel, H. (2004). Automatic keyword extraction from spoken text. a comparison of two lexical resources: the edr and wordnet. In *the 4th international language resources and evaluation, european language resource association* (p. 2205-2208).

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. In *Text mining. applications and theory* (p. 1-20). John Wiley and Sons, Ltd.

Saif, A., & Aziz, M. (2011). An automatic collocation extraction from arabic corpus. *Journal of Computer Science*, *7*(1), 6-11.

Salton, G., Yang, C., & Yu, C. (1975). A theory of term importance in automatic text analysis. *Journal of the American society for Information Science*, *26*(1), 164-168.

Sarkar, K., Nasipuri, M., & Ghose, S. (2010). A new approach to keyphrase extraction using neural networks. *IJCSI International Journal of Computer Science*, *7*(3).

Shah, M. (2008). The islamic world. In (p. 261-277). New York; London: Routledge.

Suzuki, Y., Fukumoto, F., & Sekiguchi, Y. (1998). Keyword extraction of radio news using term weighting with an encyclopedia and newspaper articles. In *the 21st annual international acm sigir conference on research and development in information retrieval* (p. 373-374).

Täckström, O., Das, D., Petrov, S., McDonald, R., & Nivre, J. (2013). Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, *1*, 1-12.

Tiwari, R., Zhang, C., & Solorio, T. (2010). A supervised machine learning approach of extracting coexpression relationship among genes from literature. In *Information reuse and integration (iri)* (p. 98 - 103).

Wartena, C., Brussee, R., & Slakhorst, W. (2010). Keyword extraction using word co-occurrence. In *Workshops on database and expert systems applications* (p. 54-58).

Witten, I., Paynter, G., Frank, E., Gutwin, C., & Nevill-Manning, C. (1999). Kea: Practical automatic keyphrase extraction. In *Proceedings of the 4th acm conference on digital library* (p. 254-255).

Zhang, K., Xu, H., Tang, J., & Li, J. (2006). Keyword extraction using support vector machine. In *Proceedings of the 7th international conference on advances in web-age information management* (p. 85-96).

Zhao, L., Yang, L., & Ma, X. (2010). Using tag to help keyword extraction. In *Computer and information application (iccia)* (p. 95-98).

Appendices

Appendix A

Dataset Statistics

The collected dataset consists of 150 Arabic documents, each document contains an Arabic abstract with its title. The used dataset has been divided randomly using a Perl program in two sets: training and testing datasets.

A training dataset that consists of 100 documents. And a set of testing that consists of 50 documents. Many statistics have been extracted from the two sets (Training and Set) and were arranged into tables.

The First statistics that were extracted from the whole documents (Abstracts and Titles) of traning and test set. Tables show in details every document name that is chosen for the training set in the first column.

The total keywords and keyphrases that are built manually for each file in the second column. The third column shows how many of these keywords and keyphrases are existed in the document (Abstract and Title). While the fourth column shows how many of these keywords and keyphrases are not existed in the document (Abstract and Title).

Finally the fifth column shows the expected recall for each document by dividing number of existed keywords and keyphrases by the total number of keywords and keyphrases in that document.

Training Set
Search Area: The Document (Abstract or Title)

Keyphrases files	Total keyphrases	Number of Existed	Number of Non Existed	Expected Recall
1.ABSTR	3	2	1	0.6667
3.ABSTR	8	4	4	0.5
4.ABSTR	4	2	2	0.5
5.ABSTR	5	4	1	0.8
8.ABSTR	2	2	0	1
11.ABSTR	4	3	1	0.75
15.ABSTR	3	2	1	0.6667
16.ABSTR	3	3	0	1
17.ABSTR	6	5	1	0.8333
19.ABSTR	4	1	3	0.25
20.ABSTR	4	3	1	0.75
22.ABSTR	3	2	1	0.6667
24.ABSTR	3	3	0	1
25.ABSTR	4	2	2	0.5
26.ABSTR	4	1	3	0.25
27.ABSTR	3	2	1	0.6667
28.ABSTR	3	1	2	0.3333
29.ABSTR	4	4	0	1
30.ABSTR	3	2	1	0.6667
31.ABSTR	4	4	0	1
32.ABSTR	12	2	10	0.1667
34.ABSTR	3	3	0	1
35.ABSTR	3	3	0	1
36.ABSTR	3	1	2	0.3333
37.ABSTR	5	3	2	0.6
38.ABSTR	3	3	0	1
39.ABSTR	3	2	1	0.6667
40.ABSTR	3	1	2	0.3333
41.ABSTR	3	2	1	0.6667
44.ABSTR	3	3	0	1
46.ABSTR	5	3	2	0.6
47.ABSTR	4	3	1	0.75

49.ABSTR	8	4	4	0.5
51.ABSTR	3	2	1	0.6667
53.ABSTR	4	2	2	0.5
54.ABSTR	4	4	0	1
55.ABSTR	3	3	0	1
56.ABSTR	3	2	1	0.6667
57.ABSTR	3	2	1	0.6667
58.ABSTR	6	3	3	0.5
59.ABSTR	10	1	9	0.1
60.ABSTR	3	2	1	0.6667
61.ABSTR	3	3	0	1
63.ABSTR	3	3	0	1
65.ABSTR	2	1	1	0.5
66.ABSTR	4	2	2	0.5
67.ABSTR	3	3	0	1
68.ABSTR	3	1	2	0.3333
70.ABSTR	4	2	2	0.5
72.ABSTR	4	3	1	0.75
74.ABSTR	3	2	1	0.6667
75.ABSTR	3	1	2	0.3333
76.ABSTR	5	2	3	0.4
77.ABSTR	5	2	3	0.4
79.ABSTR	5	2	3	0.4
81.ABSTR	8	1	7	0.125
83.ABSTR	2	2	0	1
86.ABSTR	3	3	0	1
88.ABSTR	8	6	2	0.75
89.ABSTR	4	2	2	0.5
91.ABSTR	4	4	0	1
94.ABSTR	2	1	1	0.5
95.ABSTR	3	1	2	0.3333
98.ABSTR	5	3	2	0.6
100.ABSTR	3	1	2	0.3333
101.ABSTR	4	2	2	0.5
102.ABSTR	3	2	1	0.6667
103.ABSTR	3	3	0	1
104.ABSTR	3	3	0	1
110.ABSTR	3	2	1	0.6667

111.ABSTR	4	2	2	0.5
113.ABSTR	4	2	2	0.5
115.ABSTR	3	2	1	0.6667
117.ABSTR	2	1	1	0.5
118.ABSTR	4	3	1	0.75
119.ABSTR	3	3	0	1
120.ABSTR	5	3	2	0.6
122.ABSTR	5	2	3	0.4
123.ABSTR	3	3	0	1
124.ABSTR	3	3	0	1
125.ABSTR	4	3	1	0.75
128.ABSTR	3	3	0	1
129.ABSTR	2	1	1	0.5
131.ABSTR	4	4	0	1
132.ABSTR	5	2	3	0.4
133.ABSTR	6	5	1	0.8333
134.ABSTR	5	2	3	0.4
136.ABSTR	6	5	1	0.8333
137.ABSTR	5	2	3	0.4
138.ABSTR	6	3	3	0.5
140.ABSTR	5	4	1	0.8
141.ABSTR	5	4	1	0.8
142.ABSTR	5	4	1	0.8
143.ABSTR	3	3	0	1
144.ABSTR	3	2	1	0.6667
146.ABSTR	3	3	0	1
147.ABSTR	4	2	2	0.5
148.ABSTR	4	4	0	1
149.ABSTR	5	3	2	0.6
150.ABSTR	3	1	2	0.3333
Average	**401**	**253**	**148**	**0.6697**

Test Set
Search Area: The Document (Abstract or Title)

Keyphrases files	Total keyphrases	Number of Existed ones	Number of Non Existed Ones	Expected Recall
2.UNCONTR	4	3	1	0.75
6.UNCONTR	4	1	3	0.25
7.UNCONTR	5	2	3	0.4
9.UNCONTR	4	2	2	0.5
10.UNCONTR	4	2	2	0.5
12.UNCONTR	2	2	0	1
13.UNCONTR	3	1	2	0.3333
14.UNCONTR	2	2	0	1
18.UNCONTR	3	3	0	1
21.UNCONTR	3	3	0	1
23.UNCONTR	4	1	3	0.25
33.UNCONTR	5	0	5	0
42.UNCONTR	3	2	1	0.6667
43.UNCONTR	5	3	2	0.6
45.UNCONTR	4	2	2	0.5
48.UNCONTR	6	1	5	0.1667
50.UNCONTR	4	2	2	0.5
52.UNCONTR	4	2	2	0.5
62.UNCONTR	3	1	2	0.3333
64.UNCONTR	4	4	0	1
69.UNCONTR	3	3	0	1
71.UNCONTR	5	3	2	0.6
73.UNCONTR	3	1	2	0.3333
78.UNCONTR	8	1	7	0.125
80.UNCONTR	5	2	3	0.4
82.UNCONTR	5	5	0	1
84.UNCONTR	3	1	2	0.3333
85.UNCONTR	4	4	0	1
87.UNCONTR	4	3	1	0.75

90.UNCONTR	3	3	0	1
92.UNCONTR	4	3	1	0.75
93.UNCONTR	4	3	1	0.75
96.UNCONTR	3	1	2	0.3333
97.UNCONTR	4	4	0	1
99.UNCONTR	3	3	0	1
105.UNCONTR	3	2	1	0.6667
106.UNCONTR	3	3	0	1
107.UNCONTR	4	4	0	1
108.UNCONTR	3	3	0	1
109.UNCONTR	3	3	0	1
112.UNCONTR	3	1	2	0.3333
114.UNCONTR	8	5	3	0.625
116.UNCONTR	3	3	0	1
121.UNCONTR	3	2	1	0.6667
126.UNCONTR	3	2	1	0.6667
127.UNCONTR	4	3	1	0.75
130.UNCONTR	3	3	0	1
135.UNCONTR	6	2	4	0.3333
139.UNCONTR	5	5	0	1
145.UNCONTR	4	3	1	0.75
Average	**195**	**123**	**72**	**0.6683**

Training Set
Search Area: Abstract

Keyphrases files	Total keyphrases	Number of Existed ones	Number of Non Existed Ones	Expected Recall
1.UNCONTR	3	2	1	0.6667
3.UNCONTR	8	4	4	0.5
4.UNCONTR	4	1	3	0.25
5.UNCONTR	5	2	3	0.4
8.UNCONTR	2	2	0	1
11.UNCONTR	4	3	1	0.75
15.UNCONTR	3	2	1	0.6667
16.UNCONTR	3	3	0	1
17.UNCONTR	6	4	2	0.6667
19.UNCONTR	4	1	3	0.25
20.UNCONTR	4	3	1	0.75
22.UNCONTR	3	2	1	0.6667
24.UNCONTR	3	3	0	1
25.UNCONTR	4	2	2	0.5
26.UNCONTR	4	0	4	0
27.UNCONTR	3	1	2	0.3333
28.UNCONTR	3	1	2	0.3333
29.UNCONTR	4	2	2	0.5
30.UNCONTR	3	2	1	0.6667
31.UNCONTR	4	4	0	1
32.UNCONTR	12	2	10	0.1667
34.UNCONTR	3	2	1	0.6667
35.UNCONTR	3	3	0	1
36.UNCONTR	3	1	2	0.3333
37.UNCONTR	5	2	3	0.4
38.UNCONTR	3	3	0	1
39.UNCONTR	3	2	1	0.6667
40.UNCONTR	3	0	3	0

41.UNCONTR	3	1	2	0.3333
44.UNCONTR	3	3	0	1
46.UNCONTR	5	1	4	0.2
47.UNCONTR	4	3	1	0.75
49.UNCONTR	8	2	6	0.25
51.UNCONTR	3	2	1	0.6667
53.UNCONTR	4	2	2	0.5
54.UNCONTR	4	3	1	0.75
55.UNCONTR	3	0	3	0
56.UNCONTR	3	0	3	0
57.UNCONTR	3	1	2	0.3333
58.UNCONTR	6	2	4	0.3333
59.UNCONTR	10	1	9	0.1
60.UNCONTR	3	1	2	0.3333
61.UNCONTR	3	3	0	1
63.UNCONTR	3	3	0	1
65.UNCONTR	2	1	1	0.5
66.UNCONTR	4	1	3	0.25
67.UNCONTR	3	2	1	0.6667
68.UNCONTR	3	1	2	0.3333
70.UNCONTR	4	2	2	0.5
72.UNCONTR	4	2	2	0.5
74.UNCONTR	3	2	1	0.6667
75.UNCONTR	3	1	2	0.3333
76.UNCONTR	5	2	3	0.4
77.UNCONTR	5	2	3	0.4
79.UNCONTR	5	2	3	0.4
81.UNCONTR	8	0	8	0
83.UNCONTR	2	2	0	1
86.UNCONTR	3	3	0	1
88.UNCONTR	8	6	2	0.75
89.UNCONTR	4	2	2	0.5
91.UNCONTR	4	4	0	1
94.UNCONTR	2	1	1	0.5
95.UNCONTR	3	1	2	0.3333
98.UNCONTR	5	3	2	0.6
100.UNCONTR	3	1	2	0.3333
101.UNCONTR	4	2	2	0.5

102.UNCONTR	3	2	1	0.6667
103.UNCONTR	3	3	0	1
104.UNCONTR	3	1	2	0.3333
110.UNCONTR	3	2	1	0.6667
111.UNCONTR	4	2	2	0.5
113.UNCONTR	4	2	2	0.5
115.UNCONTR	3	2	1	0.6667
117.UNCONTR	2	1	1	0.5
118.UNCONTR	4	3	1	0.75
119.UNCONTR	3	2	1	0.6667
120.UNCONTR	5	3	2	0.6
122.UNCONTR	5	2	3	0.4
123.UNCONTR	3	3	0	1
124.UNCONTR	3	3	0	1
125.UNCONTR	4	3	1	0.75
128.UNCONTR	3	3	0	1
129.UNCONTR	2	0	2	0
131.UNCONTR	4	4	0	1
132.UNCONTR	5	2	3	0.4
133.UNCONTR	6	5	1	0.8333
134.UNCONTR	5	2	3	0.4
136.UNCONTR	6	4	2	0.6667
137.UNCONTR	5	2	3	0.4
138.UNCONTR	6	3	3	0.5
140.UNCONTR	5	4	1	0.8
141.UNCONTR	5	3	2	0.6
142.UNCONTR	5	3	2	0.6
143.UNCONTR	3	3	0	1
144.UNCONTR	3	1	2	0.3333
146.UNCONTR	3	2	1	0.6667
147.UNCONTR	4	2	2	0.5
148.UNCONTR	4	3	1	0.75
149.UNCONTR	5	3	2	0.6
150.UNCONTR	3	1	2	0.3333
Average	**401**	**214**	**187**	**0.5653**

Test Set
Search Area: Abstract

Keyphrases files	**Total keyphrases**	**Number of Existed ones**	**Number of Non Existed Ones**	**Expected Recall**
2.UNCONTR	4	2	2	0.5
6.UNCONTR	4	1	3	0.25
7.UNCONTR	5	2	3	0.4
9.UNCONTR	4	2	2	0.5
10.UNCONTR	4	1	3	0.25
12.UNCONTR	2	2	0	1
13.UNCONTR	3	1	2	0.3333
14.UNCONTR	2	1	1	0.5
18.UNCONTR	3	3	0	1
21.UNCONTR	3	2	1	0.6667
23.UNCONTR	4	1	3	0.25
33.UNCONTR	5	0	5	0
42.UNCONTR	3	1	2	0.3333
43.UNCONTR	5	1	4	0.2
45.UNCONTR	4	2	2	0.5
48.UNCONTR	6	1	5	0.1667
50.UNCONTR	4	1	3	0.25
52.UNCONTR	4	1	3	0.25
62.UNCONTR	3	1	2	0.3333
64.UNCONTR	4	3	1	0.75
69.UNCONTR	3	2	1	0.6667
71.UNCONTR	5	3	2	0.6
73.UNCONTR	3	1	2	0.3333
78.UNCONTR	8	1	7	0.125
80.UNCONTR	5	2	3	0.4
82.UNCONTR	5	5	0	1

84.UNCONTR	3	0	3	0
85.UNCONTR	4	4	0	1
87.UNCONTR	4	3	1	0.75
90.UNCONTR	3	3	0	1
92.UNCONTR	4	3	1	0.75
93.UNCONTR	4	3	1	0.75
96.UNCONTR	3	1	2	0.3333
97.UNCONTR	4	3	1	0.75
99.UNCONTR	3	3	0	1
105.UNCONTR	3	2	1	0.6667
106.UNCONTR	3	3	0	1
107.UNCONTR	4	4	0	1
108.UNCONTR	3	3	0	1
109.UNCONTR	3	3	0	1
112.UNCONTR	3	1	2	0.3333
114.UNCONTR	8	5	3	0.625
116.UNCONTR	3	3	0	1
121.UNCONTR	3	2	1	0.6667
126.UNCONTR	3	2	1	0.6667
127.UNCONTR	4	3	1	0.75
130.UNCONTR	3	2	1	0.6667
135.UNCONTR	6	2	4	0.3333
139.UNCONTR	5	4	1	0.8
145.UNCONTR	4	3	1	0.75
Average	**195**	**108**	**87**	**0.5830**

Training Set
Search Area: Title

Keyphrases files	Total keyphrases	Number of Existed ones	Number of Non Existed Ones	Expected Recall
1.UNCONTR	3	1	2	0.3333
3.UNCONTR	8	2	6	0.25
4.UNCONTR	4	1	3	0.25
5.UNCONTR	5	4	1	0.8
8.UNCONTR	2	1	1	0.5
11.UNCONTR	4	0	4	0
15.UNCONTR	3	2	1	0.6667
16.UNCONTR	3	3	0	1
17.UNCONTR	6	4	2	0.6667
19.UNCONTR	4	1	3	0.25
20.UNCONTR	4	1	3	0.25
22.UNCONTR	3	2	1	0.6667
24.UNCONTR	3	2	1	0.6667
25.UNCONTR	4	2	2	0.5
26.UNCONTR	4	1	3	0.25
27.UNCONTR	3	2	1	0.6667
28.UNCONTR	3	0	3	0
29.UNCONTR	4	2	2	0.5
30.UNCONTR	3	2	1	0.6667
31.UNCONTR	4	4	0	1
32.UNCONTR	12	0	12	0
34.UNCONTR	3	2	1	0.6667
35.UNCONTR	3	3	0	1
36.UNCONTR	3	0	3	0
37.UNCONTR	5	3	2	0.6
38.UNCONTR	3	2	1	0.6667
39.UNCONTR	3	1	2	0.3333

40.UNCONTR	3	1	2	0.3333
41.UNCONTR	3	2	1	0.6667
44.UNCONTR	3	2	1	0.6667
46.UNCONTR	5	2	3	0.4
47.UNCONTR	4	3	1	0.75
49.UNCONTR	8	2	6	0.25
51.UNCONTR	3	1	2	0.3333
53.UNCONTR	4	2	2	0.5
54.UNCONTR	4	3	1	0.75
55.UNCONTR	3	3	0	1
56.UNCONTR	3	2	1	0.6667
57.UNCONTR	3	2	1	0.6667
58.UNCONTR	6	2	4	0.3333
59.UNCONTR	10	0	10	0
60.UNCONTR	3	2	1	0.6667
61.UNCONTR	3	1	2	0.3333
63.UNCONTR	3	3	0	1
65.UNCONTR	2	1	1	0.5
66.UNCONTR	4	1	3	0.25
67.UNCONTR	3	3	0	1
68.UNCONTR	3	1	2	0.3333
70.UNCONTR	4	2	2	0.5
72.UNCONTR	4	2	2	0.5
74.UNCONTR	3	0	3	0
75.UNCONTR	3	1	2	0.3333
76.UNCONTR	5	1	4	0.2
77.UNCONTR	5	2	3	0.4
79.UNCONTR	5	1	4	0.2
81.UNCONTR	8	1	7	0.125
83.UNCONTR	2	2	0	1
86.UNCONTR	3	3	0	1
88.UNCONTR	8	3	5	0.375
89.UNCONTR	4	0	4	0
91.UNCONTR	4	4	0	1
94.UNCONTR	2	1	1	0.5
95.UNCONTR	3	1	2	0.3333
98.UNCONTR	5	3	2	0.6
100.UNCONTR	3	0	3	0

101.UNCONTR	4	1	3	0.25
102.UNCONTR	3	1	2	0.3333
103.UNCONTR	3	3	0	1
104.UNCONTR	3	2	1	0.6667
110.UNCONTR	3	1	2	0.3333
111.UNCONTR	4	0	4	0
113.UNCONTR	4	1	3	0.25
115.UNCONTR	3	1	2	0.3333
117.UNCONTR	2	1	1	0.5
118.UNCONTR	4	2	2	0.5
119.UNCONTR	3	3	0	1
120.UNCONTR	5	3	2	0.6
122.UNCONTR	5	2	3	0.4
123.UNCONTR	3	2	1	0.6667
124.UNCONTR	3	2	1	0.6667
125.UNCONTR	4	3	1	0.75
128.UNCONTR	3	3	0	1
129.UNCONTR	2	1	1	0.5
131.UNCONTR	4	3	1	0.75
132.UNCONTR	5	1	4	0.2
133.UNCONTR	6	2	4	0.3333
134.UNCONTR	5	1	4	0.2
136.UNCONTR	6	1	5	0.1667
137.UNCONTR	5	2	3	0.4
138.UNCONTR	6	3	3	0.5
140.UNCONTR	5	2	3	0.4
141.UNCONTR	5	3	2	0.6
142.UNCONTR	5	2	3	0.4
143.UNCONTR	3	3	0	1
144.UNCONTR	3	2	1	0.6667
146.UNCONTR	3	3	0	1
147.UNCONTR	4	1	3	0.25
148.UNCONTR	4	3	1	0.75
149.UNCONTR	5	2	3	0.4
150.UNCONTR	3	0	3	0
Average	**401**	**180**	**221**	**0.4938**

Test Set
Search Area: Title

Keyphrases files	Total keyphrases	Number of Existed ones	Number of Non Existed Ones	Expected Recall
2.UNCONTR	4	1	3	0.25
6.UNCONTR	4	1	3	0.25
7.UNCONTR	5	1	4	0.2
9.UNCONTR	4	1	3	0.25
10.UNCONTR	4	2	2	0.5
12.UNCONTR	2	2	0	1
13.UNCONTR	3	1	2	0.3333
14.UNCONTR	2	2	0	1
18.UNCONTR	3	2	1	0.6667
21.UNCONTR	3	2	1	0.6667
23.UNCONTR	4	1	3	0.25
33.UNCONTR	5	0	5	0
42.UNCONTR	3	2	1	0.6667
43.UNCONTR	5	2	3	0.4
45.UNCONTR	4	1	3	0.25
48.UNCONTR	6	1	5	0.1667
50.UNCONTR	4	2	2	0.5
52.UNCONTR	4	2	2	0.5
62.UNCONTR	3	1	2	0.3333
64.UNCONTR	4	4	0	1
69.UNCONTR	3	3	0	1
71.UNCONTR	5	1	4	0.2
73.UNCONTR	3	0	3	0
78.UNCONTR	8	0	8	0
80.UNCONTR	5	1	4	0.2
82.UNCONTR	5	3	2	0.6
84.UNCONTR	3	1	2	0.3333
85.UNCONTR	4	3	1	0.75
87.UNCONTR	4	2	2	0.5

90.UNCONTR	3	2	1	0.6667
92.UNCONTR	4	1	3	0.25
93.UNCONTR	4	3	1	0.75
96.UNCONTR	3	0	3	0
97.UNCONTR	4	3	1	0.75
99.UNCONTR	3	2	1	0.6667
105.UNCONTR	3	1	2	0.3333
106.UNCONTR	3	2	1	0.6667
107.UNCONTR	4	4	0	1
108.UNCONTR	3	3	0	1
109.UNCONTR	3	3	0	1
112.UNCONTR	3	1	2	0.3333
114.UNCONTR	8	3	5	0.375
116.UNCONTR	3	3	0	1
121.UNCONTR	3	1	2	0.3333
126.UNCONTR	3	2	1	0.6667
127.UNCONTR	4	3	1	0.75
130.UNCONTR	3	3	0	1
135.UNCONTR	6	1	5	0.1667
139.UNCONTR	5	3	2	0.6
145.UNCONTR	4	2	2	0.5
Average	**195**	**91**	**104**	**0.5115**

Appendix B

Experiments Results

Experiments Results using Backward Graph and Window Size=2 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1143	45	0.1282	0.0404	0.0602
No	No	No	401	1095	2	0.0058	0.0017	0.0027
Yes	Yes	No	401	839	57	0.1492	0.0689	0.0922
Yes	No	Yes	401	1028	57	0.1613	0.0563	0.0811
No	Yes	No	401	861	47	0.1237	0.0562	0.0750
No	No	Yes	401	1095	1	0.0025	0.0009	0.0013
Yes	Yes	Yes	401	780	91	0.2420	0.1146	0.1513
No	Yes	Yes	401	824	82	0.2189	0.0999	0.1335

Experiments Results using Backward Graph and Window Size=3 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1147	67	0.1828	0.0574	0.0855
No	No	No	401	1088	2	0.0050	0.0019	0.0028
Yes	Yes	No	401	839	57	0.1492	0.0689	0.0922
Yes	No	Yes	401	1063	76	0.2093	0.0708	0.1024
No	Yes	No	401	861	47	0.1237	0.0562	0.0750
No	No	Yes	401	1088	3	0.0063	0.0028	0.0038
Yes	Yes	Yes	401	780	91	0.2420	0.1146	0.1513
No	Yes	Yes	401	824	82	0.2189	0.0999	0.1335

Experiments Results using Backward Graph and Window Size=5 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1150	73	0.1984	0.0643	0.0942
No	No	No	401	1078	7	0.0203	0.0067	0.0100
Yes	Yes	No	401	839	57	0.1492	0.0689	0.0922
Yes	No	Yes	401	1087	79	0.2167	0.0742	0.1068
No	Yes	No	401	861	47	0.1237	0.0562	0.0750
No	No	Yes	401	1078	8	0.0216	0.0075	0.0110
Yes	Yes	Yes	401	780	91	0.2420	0.1146	0.1513
No	Yes	Yes	401	824	82	0.2189	0.0999	0.1335

Experiments Results using Backward Graph and Window Size=10 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1154	78	0.2157	0.0689	0.1013
No	No	No	401	1051	10	0.0248	0.0119	0.0146
Yes	Yes	No	401	839	57	0.1492	0.0689	0.0922
Yes	No	Yes	401	1094	81	0.2257	0.0785	0.1105
No	Yes	No	401	861	47	0.1237	0.0562	0.0750
No	No	Yes	401	1051	11	0.0281	0.0127	0.0159
Yes	Yes	Yes	401	780	91	0.2420	0.1146	0.1513
No	Yes	Yes	401	824	82	0.2189	0.0999	0.1335

Experiments Results using Backward Graph and Window Size=2 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	580	23	0.1373	0.0392	0.0604
No	No	No	195	556	2	0.0150	0.0045	0.0069
Yes	Yes	No	195	496	40	0.2415	0.0833	0.1212
Yes	No	Yes	195	524	27	0.1620	0.0539	0.0794
No	Yes	No	195	515	27	0.1642	0.0544	0.0799
No	No	Yes	195	556	2	0.0150	0.0045	0.0069
Yes	Yes	Yes	195	408	50	0.2958	0.1295	0.1754
No	Yes	Yes	195	431	44	0.2575	0.1100	0.1506

Experiments Results using Backward Graph and Window Size=3 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	581	29	0.1713	0.0489	0.0756
No	No	No	195	550	3	0.0190	0.0065	0.0096
Yes	Yes	No	195	496	40	0.2415	0.0833	0.1212
Yes	No	Yes	195	551	31	0.1820	0.0570	0.0858
No	Yes	No	195	515	27	0.1642	0.0544	0.0799
No	No	Yes	195	550	3	0.0190	0.0065	0.0096
Yes	Yes	Yes	195	408	50	0.2958	0.1295	0.1754
No	Yes	Yes	195	431	44	0.2575	0.1100	0.1506

Experiments Results using Backward Graph and Window Size=5 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	581	35	0.2027	0.0607	0.0925
No	No	No	195	548	4	0.0165	0.0077	0.0104
Yes	Yes	No	195	496	40	0.2415	0.0833	0.1212
Yes	No	Yes	195	559	37	0.2143	0.0742	0.1051
No	Yes	No	195	515	27	0.1642	0.0544	0.0799
No	No	Yes	195	548	3	0.0140	0.0060	0.0084
Yes	Yes	Yes	195	408	50	0.2958	0.1295	0.1754
No	Yes	Yes	195	431	44	0.2575	0.1100	0.1506

Experiments Results using Backward Graph and Window Size=10 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	577	35	0.1960	0.0583	0.0893
No	No	No	195	547	6	0.0298	0.0107	0.0155
Yes	Yes	No	195	496	40	0.2415	0.0833	0.1212
Yes	No	Yes	195	565	39	0.2168	0.0665	0.1009
No	Yes	No	195	515	27	0.1642	0.0544	0.0799
No	No	Yes	195	547	6	0.0298	0.0107	0.0155
Yes	Yes	Yes	195	408	50	0.2958	0.1295	0.1754
No	Yes	Yes	195	431	44	0.2575	0.1100	0.1506

Experiments Results using Forward Graph and Window Size=2 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1141	47	0.1259	0.0396	0.0589
No	No	No	401	1074	8	0.0196	0.0071	0.0101
Yes	Yes	No	401	982	107	0.2785	0.1093	0.1535
Yes	No	Yes	401	1023	54	0.1390	0.0491	0.0707
No	Yes	No	401	1007	93	0.2448	0.0919	0.1305
No	No	Yes	401	1074	8	0.0196	0.0071	0.0101
Yes	Yes	Yes	401	684	111	0.2948	0.1936	0.2102
No	Yes	Yes	401	774	98	0.2594	0.1468	0.1700

Experiments Results using Forward Graph and Window Size=3 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1131	53	0.1464	0.0456	0.0682
No	No	No	401	1079	9	0.0236	0.0089	0.0126
Yes	Yes	No	401	987	104	0.2710	0.1055	0.1478
Yes	No	Yes	401	1045	67	0.1832	0.0647	0.0906
No	Yes	No	401	1009	101	0.2650	0.0995	0.1410
No	No	Yes	401	1079	9	0.0236	0.0089	0.0126
Yes	Yes	Yes	401	858	112	0.2935	0.1423	0.1789
No	Yes	Yes	401	942	109	0.2875	0.1215	0.1628

Experiments Results using Forward Graph and Window Size=5 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1135	63	0.1750	0.0566	0.0827
No	No	No	401	1073	10	0.0240	0.0114	0.0145
Yes	Yes	No	401	986	101	0.2662	0.1028	0.1445
Yes	No	Yes	401	1063	79	0.2132	0.0757	0.1073
No	Yes	No	401	1008	102	0.2672	0.1005	0.1427
No	No	Yes	401	1073	10	0.0232	0.0114	0.0144
Yes	Yes	Yes	401	932	114	0.3008	0.1303	0.1707
No	Yes	Yes	401	989	116	0.3018	0.1171	0.1640

Experiments Results using Forward Graph and Window Size=10 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1139	58	0.1605	0.0519	0.0754
No	No	No	401	1043	15	0.0363	0.0180	0.0209
Yes	Yes	No	401	989	103	0.2687	0.1048	0.1471
Yes	No	Yes	401	1119	74	0.1989	0.0660	0.0958
No	Yes	No	401	1021	99	0.2585	0.0974	0.1383
No	No	Yes	401	1043	15	0.0363	0.0180	0.0209
Yes	Yes	Yes	401	976	115	0.3047	0.1211	0.1672
No	Yes	Yes	401	1016	111	0.2890	0.1099	0.1552

Experiments Results using Forward Graph and Window Size=2 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	576	30	0.1625	0.0500	0.0758
No	No	No	195	562	1	0.0100	0.0040	0.0057
Yes	Yes	No	195	493	49	0.2855	0.1014	0.1473
Yes	No	Yes	195	516	30	0.1625	0.0577	0.0840
No	Yes	No	195	504	41	0.2372	0.0797	0.1174
No	No	Yes	195	562	1	0.0100	0.0040	0.0057
Yes	Yes	Yes	195	283	43	0.2465	0.1828	0.1982
No	Yes	Yes	195	324	43	0.2455	0.1611	0.1810

Experiments Results using Forward Graph and Window Size=3 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	581	24	0.1258	0.0406	0.0607
No	No	No	195	553	3	0.0190	0.0103	0.0133
Yes	Yes	No	195	493	53	0.3037	0.1097	0.1580
Yes	No	Yes	195	541	32	0.1782	0.0577	0.0860
No	Yes	No	195	512	49	0.2797	0.0978	0.1422
No	No	Yes	195	553	3	0.0190	0.0103	0.0133
Yes	Yes	Yes	195	388	55	0.3127	0.1623	0.2032
No	Yes	Yes	195	439	53	0.3010	0.1393	0.1788

Experiments Results using Forward Graph and Window Size=5 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	575	33	0.1808	0.0572	0.0858
No	No	No	195	552	7	0.0415	0.0164	0.0229
Yes	Yes	No	195	495	53	0.3073	0.1094	0.1582
Yes	No	Yes	195	543	39	0.2242	0.0740	0.1093
No	Yes	No	195	507	45	0.2590	0.0909	0.1320
No	No	Yes	195	552	7	0.0415	0.0164	0.0229
Yes	Yes	Yes	195	442	53	0.3057	0.1277	0.1734
No	Yes	Yes	195	480	49	0.2770	0.1118	0.1528

Experiments Results using Forward Graph and Window Size=10 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	583	33	0.1935	0.0561	0.0859
No	No	No	195	537	8	0.0482	0.0179	0.0258
Yes	Yes	No	195	495	47	0.2782	0.0957	0.1401
Yes	No	Yes	195	546	38	0.2268	0.0694	0.1049
No	Yes	No	195	511	43	0.2522	0.0844	0.1245
No	No	Yes	195	537	8	0.0482	0.0179	0.0258
Yes	Yes	Yes	195	467	49	0.2882	0.1077	0.1523
No	Yes	Yes	195	498	43	0.2515	0.0873	0.1267

Experiments Results using Bidirectional Graph and Window Size=2 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1132	29	0.0739	0.0267	0.0380
No	No	No	401	1142	12	0.0352	0.0110	0.0162
Yes	Yes	No	401	977	106	0.2759	0.1083	0.1511
Yes	No	Yes	401	991	47	0.1257	0.0472	0.0669
No	Yes	No	401	1005	102	0.2648	0.1009	0.1425
No	No	Yes	401	1031	28	0.0833	0.0269	0.0399
Yes	Yes	Yes	401	909	109	0.2830	0.1224	0.1640
No	Yes	Yes	401	961	109	0.2813	0.1143	0.1577

Experiments Results using Bidirectional Graph and Window Size=3 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1132	29	0.0739	0.0267	0.0380
No	No	No	401	1142	12	0.0352	0.0110	0.0162
Yes	Yes	No	401	986	113	0.2908	0.1147	0.1600
Yes	No	Yes	401	991	47	0.1257	0.0472	0.0669
No	Yes	No	401	979	105	0.2717	0.1058	0.1485
No	No	Yes	401	1031	28	0.0833	0.0269	0.0399
Yes	Yes	Yes	401	981	114	0.2920	0.1174	0.1623
No	Yes	Yes	401	976	108	0.2743	0.1097	0.1521

Experiments Results using Bidirectional Graph and Window Size=5 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1132	29	0.0739	0.0267	0.0380
No	No	No	401	1142	12	0.0352	0.0110	0.0162
Yes	Yes	No	401	983	109	0.2852	0.1116	0.1561
Yes	No	Yes	401	991	47	0.1257	0.0472	0.0669
No	Yes	No	401	1002	110	0.2877	0.1088	0.1539
No	No	Yes	401	1031	28	0.0833	0.0269	0.0399
Yes	Yes	Yes	401	980	112	0.2935	0.1142	0.1602
No	Yes	Yes	401	1001	109	0.2836	0.1072	0.1517

Experiments Results using Bidirectional Graph and Window Size=10 on Training Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	401	1132	29	0.0739	0.0267	0.0380
No	No	No	401	1142	12	0.0352	0.0110	0.0162
Yes	Yes	No	401	988	110	0.2902	0.1114	0.1568
Yes	No	Yes	401	991	47	0.1257	0.0472	0.0669
No	Yes	No	401	997	107	0.2757	0.1061	0.1495
No	No	Yes	401	1031	28	0.0833	0.0269	0.0399
Yes	Yes	Yes	401	984	118	0.3067	0.1197	0.1674
No	Yes	Yes	401	998	110	0.2795	0.1089	0.1526

Experiments Results using Bidirectional Graph and Window Size=2 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	574	14	0.0728	0.0240	0.0356
No	No	No	195	576	6	0.0317	0.0112	0.0163
Yes	Yes	No	195	493	51	0.2888	0.1062	0.1530
Yes	No	Yes	195	500	23	0.1365	0.0477	0.0697
No	Yes	No	195	507	48	0.2738	0.0964	0.1403
No	No	Yes	195	524	15	0.0788	0.0296	0.0419
Yes	Yes	Yes	195	469	56	0.3188	0.1273	0.1786
No	Yes	Yes	195	482	50	0.2882	0.1092	0.1555

Experiments Results using Bidirectional Graph and Window Size=3 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	574	14	0.0728	0.0240	0.0356
No	No	No	195	576	6	0.0317	0.0112	0.0163
Yes	Yes	No	195	498	51	0.2957	0.1040	0.1506
Yes	No	Yes	195	500	23	0.1365	0.0477	0.0697
No	Yes	No	195	508	48	0.2773	0.0993	0.1422
No	No	Yes	195	524	15	0.0788	0.0296	0.0419
Yes	Yes	Yes	195	498	55	0.3207	0.1142	0.1648
No	Yes	Yes	195	508	50	0.2890	0.1029	0.1477

Experiments Results using Bidirectional Graph and Window Size=5 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	574	14	0.0728	0.0240	0.0356
No	No	No	195	576	6	0.0317	0.0112	0.0163
Yes	Yes	No	195	497	50	0.2925	0.1012	0.1479
Yes	No	Yes	195	500	23	0.1365	0.0477	0.0697
No	Yes	No	195	509	46	0.2648	0.0923	0.1345
No	No	Yes	195	524	15	0.0788	0.0296	0.0419
Yes	Yes	Yes	195	496	56	0.3255	0.1149	0.1671
No	Yes	Yes	195	508	46	0.2648	0.0927	0.1350

Experiments Results using Bidirectional Graph and Window Size=10 on Test Set

Stopwords Removal	Applying Tagging	Using Weights	Relevant	Retrieved	Retrieved Relevants	Recall	Precision	F-measure
Yes	No	No	195	574	14	0.0728	0.0240	0.0356
No	No	No	195	576	6	0.0317	0.0112	0.0163
Yes	Yes	No	195	497	49	0.2858	0.1005	0.1462
Yes	No	Yes	195	500	23	0.1365	0.0477	0.0697
No	Yes	No	195	508	44	0.2568	0.0885	0.1295
No	No	Yes	195	524	15	0.0788	0.0296	0.0419
Yes	Yes	Yes	195	496	53	0.3098	0.1096	0.1593
No	Yes	Yes	195	507	47	0.2708	0.0947	0.1380